# Final Year Project Report

**Crime-safety App**

**Project Team**

**Akbar Zeb Khan 2012236**

Submitted in the partial fulfillment of the requirements for

The degree of Bachelor of Science in Computer Science in

the Faculty of Computing and Engineer in Sciences

Contents

[Final Year Project Report 1](#_Toc156806855)

[**Project Proposal** 1](#_Toc156806856)

[Supervisor 1](#_Toc156806857)

[Submitted by 1](#_Toc156806858)

[1. Introduction 1](#_Toc156806859)

[2. Objective 1](#_Toc156806860)

[3. Problem Description 1](#_Toc156806861)

[4. Methodology 2](#_Toc156806862)

[5. Project Scope 3](#_Toc156806863)

[6. Feasibility Study 3](#_Toc156806864)

[7. Solution Application Areas 3](#_Toc156806865)

[8. Tools/Technology 4](#_Toc156806866)

[9. Expertise of the Team Members 5](#_Toc156806867)

[10. Milestones 5](#_Toc156806868)

[11. Project Schedule 6](#_Toc156806869)

[12. Work Breakdown Structure 9](#_Toc156806870)

[13. References 10](#_Toc156806872)

[1. Introduction 1](#_Toc156806873)

[1.1. Purpose 1](#_Toc156806874)

[1.2 Document Conventions 1](#_Toc156806875)

[1.3 Intended Audience and Reading Suggestions. 1](#_Toc156806876)

[1.4 Product Scope 1](#_Toc156806877)

[1.5 References 1](#_Toc156806878)

[2. Overall Description 2](#_Toc156806879)

[2.1 Product Perspective 2](#_Toc156806880)

[2.2 Product Functions 3](#_Toc156806881)

[2.3 User Classes and Characteristics 3](#_Toc156806882)

[2.3 Operating Environment 3](#_Toc156806883)

[2.5 Design and Implementation Constraints 3](#_Toc156806884)

[2.4 User Documentation 4](#_Toc156806885)

[2.5 Assumptions and Dependencies 4](#_Toc156806886)

[3. External Interface Requirements 4](#_Toc156806887)

[3.1 User Interfaces 4](#_Toc156806888)

[3.2 Hardware Interfaces 4](#_Toc156806889)

[3.3 Software Interfaces 5](#_Toc156806890)

[3.4 Communications Interfaces 5](#_Toc156806891)

[4. System Features 6](#_Toc156806892)

[4.1 System Feature 1 6](#_Toc156806893)

[4.2 System Feature 2 7](#_Toc156806894)

[4.3 System Feature 3 8](#_Toc156806895)

[4.4 System Feature 4 9](#_Toc156806896)

[4.5 System Feature 5 10](#_Toc156806897)

[4.5.1 System Feature 5.1 11](#_Toc156806898)

[4.5.2 System Feature 5.2 12](#_Toc156806899)

[4.5.3 System Feature 5.3 13](#_Toc156806900)

[4.5.4 System Feature 5.4 14](#_Toc156806901)

[4.6 System Feature 6 15](#_Toc156806902)

[4.6.1 System Feature 6.1 16](#_Toc156806903)

[4.6.2 System Feature 6.2 17](#_Toc156806904)

[4.6.3 System Feature 6.3 18](#_Toc156806905)

[4.6.4 System Feature 6.4 19](#_Toc156806906)

[4.7 System Feature 7 20](#_Toc156806907)

[4.8 System Feature 8 21](#_Toc156806908)

[4.9 System Feature 9 22](#_Toc156806909)

[4.10 System Feature 10 13](#_Toc156806910)

[4.11 System Feature 11 14](#_Toc156806911)

[4.12 System Feature 12 15](#_Toc156806912)

[5. Other Nonfunctional Requirements 17](#_Toc156806913)

[5.1 Performance Requirements 17](#_Toc156806914)

[5.1 Safety Requirements 17](#_Toc156806915)

[5.3 Security Requirements 17](#_Toc156806916)

[1.1 Software Quality Attributes 17](#_Toc156806917)

[5.5 Business Rules 17](#_Toc156806918)

[6. Other Requirements 18](#_Toc156806919)

[3.1. Section Overview 25](#_Toc156806920)

[3.2. Component n Detail (include a sub-section for each component) 25](#_Toc156806921)

[4. User Interface Design 38](#_Toc156806922)

[4.1. Section Overview 38](#_Toc156806923)

[4.2. Interface Design Rules 38](#_Toc156806924)

[4.3. GUI Components 38](#_Toc156806925)

[4.4. Detailed Description 38](#_Toc156806926)

[ Class Diagram 62](#_Toc156806928)

[ Analysis Diagram: 63](#_Toc156806929)

[ User Manual: 99](#_Toc156806930)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**Crime-Safety - Safety at Your**

**Fingertips**

### **Project Proposal**

## Submitted by

Akbar Zaib Khan

{2012236 }

**Faculty of Computing and Engineering Sciences**

[17th JUNE 2023]

# Introduction

Our project aims to develop a crime reporting and prevention mobile application that utilizes real-time updates and a comprehensive crime database. Users can report crimes, share evidence, and receive location-based alerts. Safe Karachi app enhances community safety, provides a color-coded crime map,SOS alert and enables effective communication between users, community and law enforcement agencies.

# Objective

To design a crime prevention and citizens safety application where users can report crime occurred to them along with supporting evidence, the location where crime occurred will be shown on the map as a blip. Other users in the area will get real-time notification of crime with details. Safe Karachi app will provide a color-coded safety meter indicating the crime rate in different areas (red for high crime, yellow for moderate, green for low), along with a no-go area feature for active police operations, terror situations, or natural disasters. Safe Karachi app will also enable users to report kidnappings, displaying the victim's information and image on the map where the victim was last seen, and allow police to access real-time crime locations and increase surveillance. There will be SOS and Emergency services option as well as user feed for community engagement.

# Problem Description

The problem we aim to address is the need for an efficient and accessible system to report crimes, share information, and enhance community safety, specifically in the context of Karachi, Pakistan. Karachi is known for its high crime rates, including robberies and kidnappings.

The current methods of crime reporting often suffer from delays, limited reach, and lack of real-time updates. Not everyone actively uses social media or watches news, which leads to inadequate awareness and response. Traditional channels of communication between citizens and law enforcement can be slow and inefficient. The arrival time for emergency services is not fast enough for in time rescue due to inadequate communication or information from the system .

Safe Karachi seeks to provide a solution to overcome these challenges by allowing users to report crimes, share evidence, and provide the exact location of incidents, we enable a more effective and immediate response. The app will send real-time updates and alerts to users who have their location set to a particular area, ensuring that people in the vicinity are informed promptly about a crime or an emergency. The user will have an option for SOS and emergency services which will be quick and in time services.

The app's features include a comprehensive crime database, a color-coded crime map indicating areas with high, moderate, and low crime rates, and a no-go area feature for active police operations, terror situations, or natural disasters. Real-time map blip for kidnapping victims. This information allows users to make informed decisions about their safety and helps law enforcement agencies prioritize their efforts.

Designing such a system requires addressing privacy and security concerns, as well as implementing mechanisms to verify the authenticity of reported incidents. By incorporating verified/unverified tag uploading, mandatory CNIC information and monitoring posts, we aim to discourage the posting of fake crimes, ensuring the reliability and credibility of the information shared.

While summarizing all the above, Safe Karachi app aims to address the problem of limited and delayed crime reporting by developing a crime reporting and prevention app. By utilizing real-time updates, location-based alerts, and a comprehensive crime database, we aim to enhance community safety, improve communication with law enforcement, and empower individuals to actively contribute to crime prevention efforts.

# Methodology

Our approach to tackling the problems includes the following steps:

User Interface Design: Create a user-friendly mobile application interface for easy signup and login, crime reporting, evidence sharing, and access to real-time updates and alerts. React native along with CSS frameworks will be used.

Backend Development: Utilize Node.js and MySQL for server-side logic, API endpoints.

Crime Reporting and Verification: Implement a verification process for authenticating reported crimes, including the option to upload supporting evidence for verification(verified/unverified tag) and CNIC requirement for identification.

Real-time Updates and Location-based Alerts: Send real-time updates and notifications to users in the reported crime's vicinity using location-based alerts. Real-time communication with Socket.IO.

Color-coded Crime Map : Develop a color-coded crime map to visually represent areas with high, moderate, and low crime rates.Google Map API.

Security and Privacy: Prioritize user data security through HTTPS encryption, user authentication, and secure data storage. Moderate user-generated content to prevent misuse.

Integration with Law Enforcement: Provide law enforcement agencies access to real-time crime locations and high crime areas for improved decision-making and resource allocation.

Throughout the implementation, we may utilize libraries like React native, Node.js, Socket.io, Express and Google Maps API. We will follow agile development methodologies, ensuring iterative development and continuous testing.

# Project Scope

While we aim to provide crime rates and color-coded areas on the map, conducting an in-depth crime analysis, such as trend identification or predictive modeling, is beyond the scope of this project. The progression of the system relies on the presumption of users having a dependable internet connection to access live updates, submit reports, and receive alerts. Additionally, we assume that users will have smartphones or devices capable of running the app and that law enforcement agencies will collaborate in utilizing the app for real-time crime information.

# Feasibility Study

**i. Risks Involved:**

Technical Risks: Challenges in integrating real-time data updates and ensuring the accuracy of crime reports may surface. Robust testing and a verification system will be implemented to prevent these risks.

Security Risks: Risks associated with data breaches or unauthorized access to user information must be addressed. Secure coding practices, encryption protocols, and regular security audits will be implemented to mitigate these risks.

**ii. Resource Requirements:**

Computing Resources: Computers or laptops meeting the minimum requirements for development tools (e.g., Node.js, React native, socket.io and MySQL) will be needed.

Development Tools: IDEs like Visual Studio Code, and version control systems like Git, will be utilized.

Server and Hosting: A server or cloud hosting service, such as AWS, will be required for backend deployment.

Database: A MySQL database will be utilized for data storage.

# Solution Application Areas

Safe Karachi, the crime reporting and prevention app for Karachi, Pakistan, holds real value and targets the domain of public safety and law enforcement. The application can benefit various stakeholders within this domain:

Citizens: The app empowers citizens by providing them with a user-friendly platform to report crimes, share evidences related to the incidents, view color-coded map for safety and stay updated on real-time crime incidents in their vicinity. This enhances their personal safety and allows them to make informed decisions about their movements and activities.

Law Enforcement Agencies: The app enables law enforcement agencies to access real-time crime data and locations. This information can help them allocate resources effectively, prioritize their efforts, and enhance situational awareness. By receiving immediate updates and alerts, law enforcement can respond more swiftly to incidents and improve their crime prevention strategies.

Missing person update: The app enables a user to post about someone missing, the map will show a unique blip that includes missing person’s details, pictures and the exact location where the person was last seen. Anyone can view the details and help locate the person.

Community Safety Organizations: The app can be utilized by non-profit organizations or community safety groups to co-operate with law enforcement agencies and make efforts in preventing crime. The app can facilitate information sharing, community engagement, and targeted interventions in areas with high crime rates.

Businesses and Tourism Industry: The app's crime map feature, which color-codes areas based on crime rates, can benefit businesses and the tourism industry. They can use this information to make informed decisions about site selection, route planning, and resource allocation. It contributes to creating a safer environment for businesses to operate and attracts more tourists to the city.

Emergency Services: The app's functionality to report emergencies and share live locations can assist emergency services, such as ambulances and fire departments, in responding quickly to incidents. It helps optimize their response times and potentially saves lives.

# Tools/Technology

**Hardware**

Computers or laptops: For development and testing purposes.

Mobile devices: To test the app on different platforms and screen sizes.

**Software**

Programming Languages: JavaScript, CSS and it’s Frameworks.

Front-end Development: React native framework for building the user interface.

Back-end Development: Node.js for server-side scripting and handling API requests.

Database: MySQL for storing crime data and user information.

Web Servers: Express.js to handle HTTP requests and serve the application.

Real-time Updates: Socket.io for real-time communication between the server and clients.

Map Integration: Google Maps API (geo-fencing)for displaying crime locations and generating maps.

Version Control: Git for managing source code versions and collaboration.

# Expertise of the Team Members

Both of the team member has good JavaScript skills. We also have intermediate knowledge of React native. Further technologies will be learnt in summer break. The project is of equal interest to both members.

# Milestones

**Research and Requirements Gathering:**

Conduct research on crime statistics and reporting systems in Karachi.

Define project requirements and scope.

Create a context diagram to identify the key actors and their goals.

**User Interface Design:**

Design the user interface for Safe Karachi app.

Gather feedback from users and stakeholders.

**Backend Development:**

Set up the server infrastructure using Node.js and Express.js.

Develop APIs for user registration, crime reporting, and location tracking.

Implement a database (e.g MySQL) to store crime data, user information, admin information and law inforcement details.

**Frontend Development:**

Develop Front-end using React native.

**Implement the user interface using React native:**

Integrate map functionality for crime visualization.

Develop features for crime reporting, user authentication, and real-time updates.

**Crime Database and Mapping**:

Design a database schema for storing crime data.

Implement algorithms to calculate crime rates and visualize them on the map.

**Authentication and Security:**

Implement user authentication and authorization mechanisms.

Ensure data security and protection of sensitive information.

**Testing and Quality Assurance:**

Conduct unit testing for each component of the app.

Perform integration testing to ensure communication between frontend and backend.

Conduct usability testing to gather feedback for improvements.

**Deployment and Documentation:**

Prepare the app for deployment on a server.

Document the installation and configuration process.

Create user documentation and guidelines for app usage.

**Evaluation and Feedback:**

Evaluate the app's performance, user experience, and effectiveness in crime prevention.

Gather feedback from users and stakeholders for further improvements.

**Finalization and Presentation:**

Prepare the final project report and documentation.

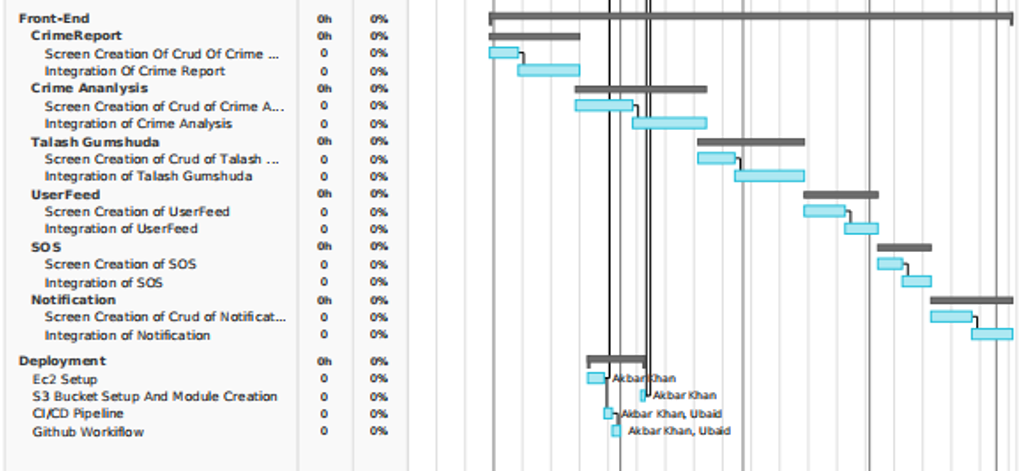
Create a presentation to showcase the app's features and accomplishments.

Present the project to the FYP panel.

1. Project Schedule

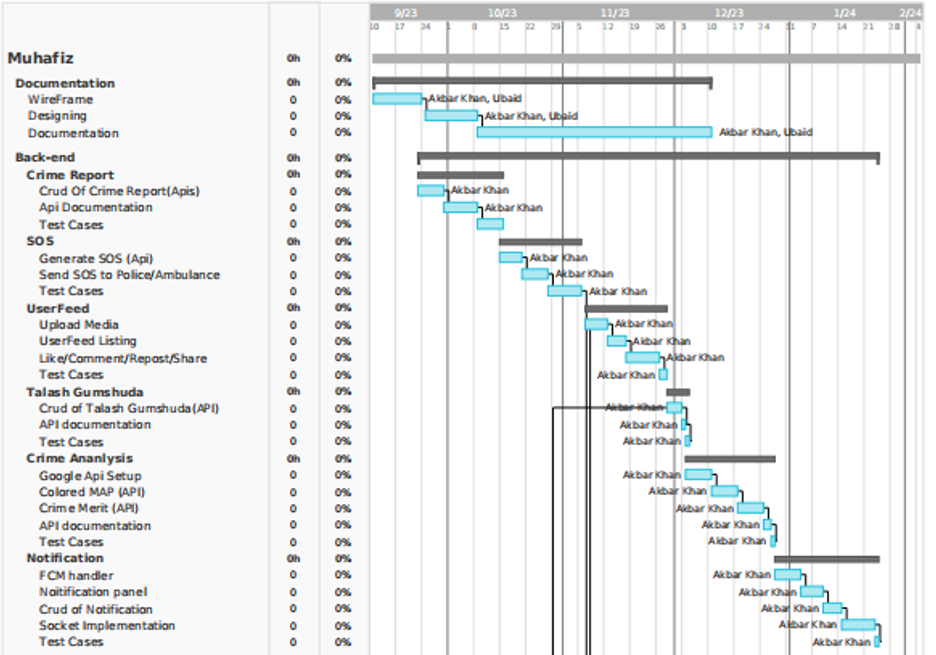
FYP 1/FYP 2:





# 12. Work Breakdown Structure

FYP 1/FYP 2:



# 

# 13. References

[1] <http://ijcsn.org/IJCSN-2019/8-3/C-App-A-Mobile-App-for-Crime-Report.pdf>

[2] https://papers.ssrn.com/sol3/papers.cfm?abstract\_id=3689762

[3] <https://www.mdpi.com/1424-8220/23/9/4350>

[4] <https://www.researchgate.net/publication/349678250_Android_Application_for_Crime_Prevention_with_GPS_Integrated_Technology>

[5] <https://www.degruyter.com/document/doi/10.1515/jisys-2022-0034>

[6][https://citizen.com/](Ubed%20and%20Akbar%20proposal%20ppt%20(1).pptx)

Software Requirements Specification

for

Crime-Safety

Prepared by

Akbar Zaib Khan - 2012236

22nd November ‘2023

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

Crime-Safety is a dedicated mobile application focused on enhancing citizen safety and providing real-time updates on various areas in Karachi. Users will have the ability to report incidents and swiftly alert others in their vicinity about any concerning events. The app features a user-friendly color-coded map displaying real-time events across the city, allowing users to stay informed about what's happening around them. Additionally, Crime-Safety offers a social feed where users can interact with one another, sharing safety tips, experiences, and information. For emergency situations, the application includes a crucial SOS function, ensuring quick access to help when needed.

## 1.2 Document Conventions

The font style for heading 1 will be Times and font size will be 18, the font style for heading 2 will be Times and font size will be 14, while the body style and size will be Arial and 11 respectively.

## 1.3 Intended Audience and Reading Suggestions.

Crime-Safety is designed to cater to a diverse audience, including the people of Karachi, tourism companies, foreign tourists, security firms, law enforcement, and emergency services. It aims to provide Karachi's residents with information to enhance their safety and well-being, while also supporting tourism companies in offering secure and enjoyable experiences for visitors. Foreign tourists can benefit from valuable safety tips and local insights, and security companies can collaborate and share their expertise. Law enforcement and emergency services can use the platform for communication, and local businesses can attract more customers and investors in a safer environment. Educational institutions can promote safety awareness, media can source reliable information, and community organizations can address safety issues at the grassroots level. Crime-Saferty serves as a comprehensive hub for improving safety and security in Karachi for all its stakeholders.

## 1.4 Product Scope

The vision for the application is to create a platform for citizen journalism and community engagement, enabling users to share real-time updates with their fellow citizens and foster a social network focused on the safety and well-being of the people of Karachi. This all-in-one solution addresses a wide range of issues faced by Karachi's residents, including crime, road blockages, riots, protests, extreme weather conditions like heavy rains, natural disasters, or acts of terrorism. Users can actively contribute by posting reports in real-time, participating in community interactions, accessing a color-coded map for situational awareness, generating SOS alerts for urgent situations, and raising awareness about missing persons. This application aims to empower citizens with the tools and information needed to respond effectively to challenges and promote the safety and prosperity of the Karachi community.

## 1.5 References

<https://www.researchgate.net/publication/262398198_Mobile_applications_for_incident_reporting_systems_in_urban_contexts_Lessons_learned_from_an_empirical_studys>

<https://dl.acm.org/doi/abs/10.1145/3290607.3312781>

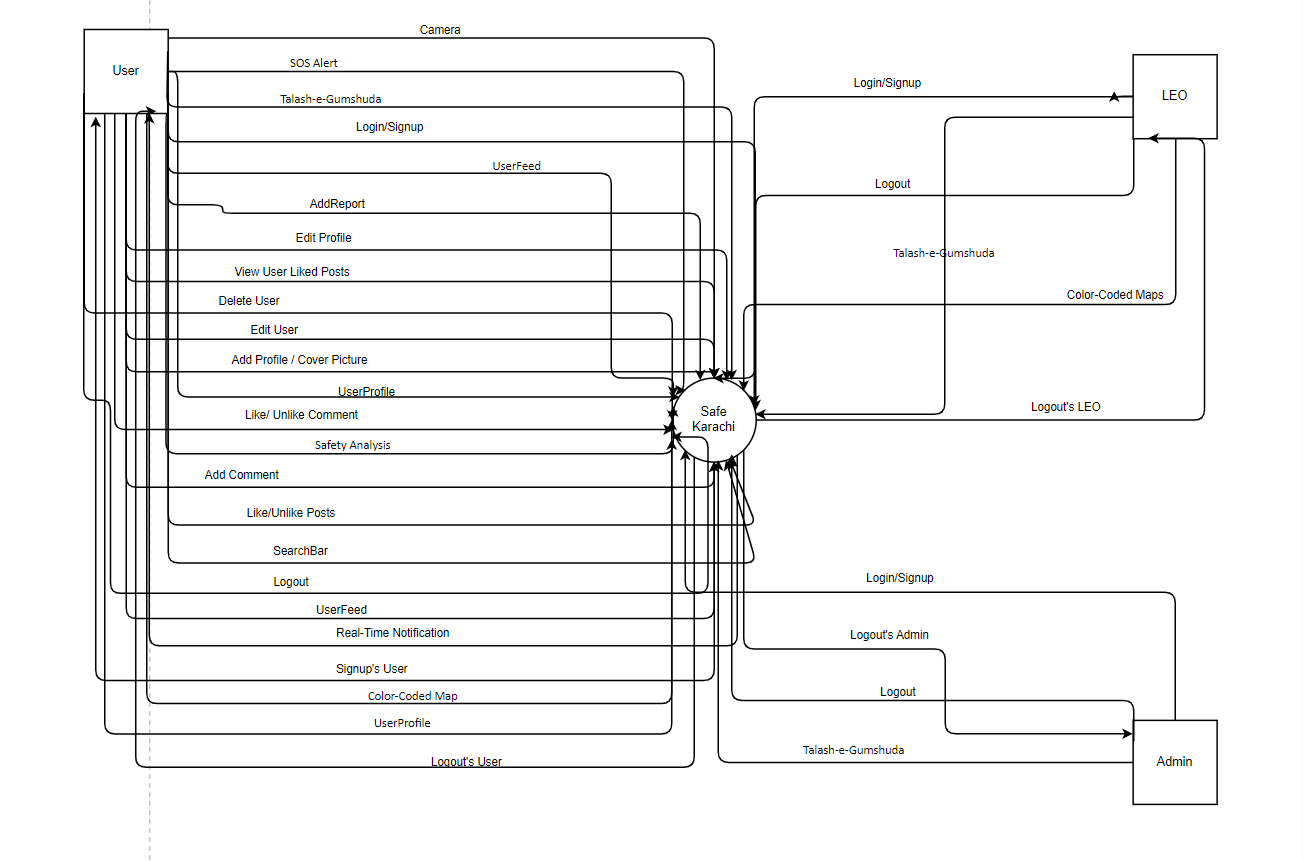
<https://bmcpublichealth.biomedcentral.com/articles/10.1186/s12889-022-13551-9>

<https://citizen.com>

# Overall Description

## 2.1 Product Perspective

Crime-Safety is a standalone mobile application being created to allow users to report and receive information about any event in real-time. It is an independent solution and not a part of a larger product family. While there may be existing apps related to public safety, our goal is to centralize the platform for a all-in-one community engagement and public safety solution for citizens of Karachi.



## 2.2 Product Functions

Main features of our application are:

1. Color-Coded Map
2. Add Report
3. Camera
4. UserFeed
   1. SearchBar
   2. Like/Unlike Post
   3. Add Comment
   4. Like/Unlike Comment
5. UserProfile
   1. Add Profile/Cover Picture
   2. Edit User Posts
   3. Delete User Post
   4. View User Liked Posts
6. Edit Profile
7. SOS Alert
8. Talash-e-Gumshuda
9. Safety Analysis
10. Real-time Notifications

## 2.3 User Classes and Characteristics

This application will be used by users for reporting any incident of public safety, viewing UserFeed, generating SOS alerts and listing missing person. A user can benefit from the user-friendly interface and time-efficient updates of the situation of city. Users should have a basic knowledge of smart phone applications. Law enforcement officers can view all the reports posted by the users and verify the authenticity of the particular posts. The law enforcement officers should have experience in using law enforcement software and tools. An administrator will manage the application affairs, manage user accounts, monitor app usage, and maintain data integrity.

## Operating Environment

The application is intended for use on mobile devices, specifically smartphones with Android and iOS operating systems.

**Android Devices:**

Operating System: Android versions 7.0 (Nougat) and later.

Development Tool: Android Studio (22.3.1).

**iOS Devices:**

Operating System: iOS versions 11 and later.

Development Tool: Xcode (latest stable version) for iOS app development.

## 2.5 Design and Implementation Constraints

* The application must comply with all the local and federal laws and regulations of the country
* Developed using React native, Node, Express and MySql
* Compatible with Android and IOS Operating systems with different hardware and under various networks
* User-friendly interface for users, admin and Law enforcement officials
* Data security and encryption for users

## User Documentation

1. SRS
2. SDS
3. STS

## Assumptions and Dependencies

* Inadequate smartphone application knowledge could hinder a user’s experience of this efficient public safety application
* We assume that a user will have a working internet connection, required operating system and mobile device.
* All users should have a valid Email and password when logging in.
* The records must be stored in a database so they are accessible.
* We assume that the application will comply to all relevant laws and regulations regarding data privacy.

# External Interface Requirements

## 3.1 User Interfaces

The system provides an elegant mobile GUI for the user, law enforcement officials and administrator. The user interface will be designed to be user-friendly and modern. Key characteristics of the mobile app user interface include:

**Landing Screen:** Upon launching the app, users will be presented with a Landing screen featuring a prominent "Get Started" button. This screen is designed to be visually appealing and welcoming to encourage user engagement along with application tutorial.

**Login/Sign up:** Login and sign up will have a modern and elegant UI. Different type of text fields and buttons will be used along with external authentication.

**Map Display:** A significant portion of the interface is dedicated to a Google Map, where users can view incident markers, area wise color-coding for safe travelling, missing person markers, and other location-based information. User-friendly zoom and distance features are available.

**User Feed:** The User Feed section displays posts in a social media feed style, with options to like and comment on Posts. Posts may include images, text, videos and other supported evidences.

**SOS Alert:** The SOS Alert interface is designed to provide quick access to emergency services with one press user-friendly and elegant buttons.

**Missing Person Listings:** The Missing Person section presents a categorized list of missing persons. Users can expand individual listings to view more details and location markers on the map.

**User Profile:** User profile will display an elegant UI for user’s profile information,user’s profile and cover pictures, user’s post and likes history.

## 3.2 Hardware Interfaces

Users can interact with the application through Android and Apple smartphones, Tablets and other handheld mobile devices with supported operating systems.

## 3.3 Software Interfaces

The application will be developed using:

* React Native
* Node.js
* Express.js
* MySQL
* Android Studio
* Google Map API
* REDUX
* REST API

## 3.4 Communications Interfaces

HTTPS communication protocol will be used for secure data transfer between app and external server.

# System Features

## 4.1 System Feature 1

|  |  |  |
| --- | --- | --- |
| **Name of Use Case:** | 1. Signup | |
| **Summary:** | This use case describes how a User registers into Crime-Safety application. | |
| **Actors:** | User | |
| **Preconditions:** | 1. Must be a new User 2. Must be on Sign up screen | |
|  | **Actor** | **Application** |
| **Basic course/Happy**  **path:** |  | 1. Application requests that the actors enter their email-ID ,password, first-name, last-name ,phone number and date of birth |
| 1. The actors enters their email-ID ,password, first-name, last-name, phone number and date of birth |  |
|  | 1. Application validates information provided by the actor |
|  | 1. Application securely stores user registration information into the user database. |
| **Alternative path:** | 7. In step 2, if the actor enters an invalid ID and/or email-ID, password, first-name, and last-name the system displays an error message. The actors can then choose to either re- enter details or cancel the Signup, at which point the use case ends.  8. In step 2, if the actor enter wrong phone number the application won’t verify the user.  9. In Step 2, if the actor is below 16 (minimum) years age , application will display an error  10 In Step 2, if the actor enters password that contains less than 8 character, no Upper case and a symbol then password won’t be valid and error would be shown. | |
| **Post condition:** | 1. Upon successful Signup, the actor will be directed to login page to log in. If not the application state is unchanged. | |
| **Author name:** | Akbar zeb khan, | |

## 4.2 System Feature 2

|  |  |  |
| --- | --- | --- |
| **Name of Use Case:** | 2. Login | |
| **Summary:** | This use case describes how a User, Admin and LEO logs into the Crime-Safety app. | |
| **Actors:** | User, Admin and LEO | |
| **Preconditions:** | 1. Must be admin or LEO, or a existing User 2. Must be on login screen | |
|  | **Actor** | **Application** |
| **Basic course/Happy**  **path:** |  | 1. Application presents options for in app login or Google login. |
| 1. The actor enter Email and password for either option. |  |
|  | 1. Application validates the entered email password and logs the actors into the application. |
| **Alternative path:** | 6. In step 2, if the actor enters an invalid Email and/or password, the application displays an invalid details error message. The actors can then choose to either re-enter login details or cancel the login, at which point the use case ends.  7. In step 2, if the actor forgot their password they can choose to reset their password by pressing on forgot password option. | |
| **Post condition:** | 1. Upon successful login, the actors can now log into the application. If not the application state is unchanged. | |
| **Author name:** | Akbar Zeb Khan | |

|  |  |  |
| --- | --- | --- |
| **Name of Use Case:** | 3. Add Report | |
| **Summary:** | This use case is about adding a report to application. | |
| **Actors:** | User | |
| **Preconditions:** | 1. The user has to be already logged in. | |
|  | **Actor** | **Application** |
|  |  | 1. Application presents a form for the actor to input detailed information about the incident. |
| **Basic course/Happy**  **path:** | 1. Actor enters incident details, including incident type, location marker on map, description, along with attached images or other media to the report. |  |
|  | 1. Application validates the report, ensuring that all necessary details are provided. |
|  | 1. Application sends actor confirmation of the successful creation of their report. |
|  | 1. The application sends alert to the particular actor and others in vicinity. |
|  | 1. Application securely stores user report information into the userFeed database. |
|  | 1. The application posts the data to user feed. |
| **Alternative path:** | 1. If the actor doesn’t fill the title, date/time, description, incident type, location and media (for evidence) then the application shows an error and the post won’t be posted. | |
| **Post condition:** | 1. Upon success, the post will be shown in the userFeed and saved to the database. | |
| **Author name:** | Akbar Zeb Khan | |

## 4.3 System Feature 3

|  |  |  |
| --- | --- | --- |
| **Name of Use Case:** | 4. Camera | |
| **Summary:** | This use case is about capturing an image or recording a video through Camera. | |
| **Actors:** | User | |
| **Preconditions:** | 1. The user has to be already logged in. | |
|  | **Actor** | **Application** |
|  | 1. Actor presses on camera icon in Dashboard page |  |
|  |  | 1. Application opens a camera. |
| **Basic course/Happy**  **path:** | 1. Actor captures image/ records video. |  |
|  | 1. Application previews image/video. |
| 1. Actor saves image/video |  |
|  | 1. Application saves image/video to device gallery and shows image/video in Add report form. |
| **Alternative path:** | 1. In Step 4, if the actor discards image/video then camera goes back to film mode again. 2. In Step 4, actor can turn on the flashlight or switch the camera from back view to front view | |
| **Post condition:** | 1. Upon success, the media will be saved in gallery and shown in add report form. | |
| **Author name:** | Akbar Zeb Khan | |

## 4.4 System Feature 4

## 4.5 System Feature 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | 5. UserFeed | | | |
| **Summary:** | This use case is about UserFeed | | | |
| **Actors:** | User | | | |
| **Preconditions:** | 1. User must be already logged in to the application. | | | |
| **Basic course/Happy**  **path:** |  | **Actor** | **Application** |  |
| 1. Actor selects the User Feed option from the app's Bottom Bar. |  |  |
|  | The application displays a feed of incident reports that contain details, media and other relevant information |  |
| 1. Actor interacts with a particular post |  |  |
|  | 1. The application displays the particular post selected by the user along with comments and verified/unverified tag. |  |
| **Alternative path:** | 1. In step 1, if the actor cannot access the userFeed page then, they will directed with a message not available and will returned back to the dashboard | | | |
| **Post condition:** | After viewing the userFeed, actor can return back to dashboard | | | |
| **Author name:** | Akbar Zeb Khan | | | |

## 4.5.1 System Feature 5.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | 5.1 SearchBar | | | |
| **Summary:** | This use case is about UserFeed SearchBar | | | |
| **Actors:** | User | | | |
| **Preconditions:** | 1. User must be already logged in to the application. 2. User must be in UserFeed | | | |
| **Basic course/Happy**  **path:** |  | **Actor** | **Application** |  |
| 1. Actor searches the post by post title, post description and post type. |  |  |
|  | 1. The application displays posts as filtered by user |  |
| **Alternative path:** | 1. In step 2, if post isn’t found in database then the application will display post not found | | | |
| **Post condition:** | 1. After searching the userFeed, actor can return back to dashboard | | | |
| **Author name:** | Akbar Zeb Khan | | | |

## 4.5.2 System Feature 5.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | 5.2 Like/Unlike Post | | | |
| **Summary:** | This use case is about liking/unliking post in the userFeed | | | |
| **Actors:** | User | | | |
| **Preconditions:** | 1. User must be already logged in to the application. 2. User must be in UserFeed | | | |
| **Basic course/Happy**  **path:** |  | **Actor** | **Application** |  |
| 1. Actor likes a particular post. |  |  |
|  | 1. The application shows new like count on the particular post liked. |  |
|  |  | 1. Actor unlikes a particular post. |  |  |
|  |  |  | 1. The application shows new like count on the particular post liked. |  |
| **Alternative path:** | 1. In step 1 and 3, even if the actor repeatedly presses the like/unlike button, only 1 like or unlike per post will be recorded per user | | | |
| **Post condition:** | 1. After liking/unliking post in the userFeed, actor can return back to dashboard | | | |
| **Author name:** | Akbar Zeb Khan | | | |

## 4.5.3 System Feature 5.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | 5.3 Add Comment | | | |
| **Summary:** | This use case is about adding comment on a post in the userFeed | | | |
| **Actors:** | User | | | |
| **Preconditions:** | 1. User must be already logged in to the application. 2. User must be in UserFeed. | | | |
| **Basic course/Happy**  **path:** |  | **Actor** | **Application** |  |
| 1. Actor presses comment icon on any post. |  |  |
|  | 1. The application displays comments of post and option to add a comment. |  |
|  |  | 1. Actor adds a comment on a post. |  |  |
|  |  |  | 1. The Application securely stores user comment into the comment database. |  |
|  |  |  | 1. The Application displays user comment on the post. |  |
| **Alternative path:** | None | | | |
| **Post condition:** | 1. After commenting on the post in the userFeed, actor can close comment sheet and return to userFeed. | | | |
| **Author name:** | Akbar Zeb Khan | | | |

## 4.5.4 System Feature 5.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | 5.4 Like/Unlike comment | | | |
| **Summary:** | This use case is about liking/unliking comments in the userFeed | | | |
| **Actors:** | User | | | |
| **Preconditions:** | 1. User must be already logged in to the application. 2. User must be in UserFeed 3. User must be in comment sheet of a post | | | |
| **Basic course/Happy**  **path:** |  | **Actor** | **Application** |  |
| 1. Actor likes a particular comment on a post. |  |  |
|  | 1. The application shows new like count on the particular comment liked. |  |
|  |  | 1. Actor unlikes a particular comment on a post. |  |  |
|  |  |  | 1. The application shows new like count on the particular comment unliked. |  |
| **Alternative path:** | 1. In step 1 and 3, even if the actor repeatedly presses the like/unlike button, only 1 like or unlike per comment will be recorded per user | | | |
| **Post condition:** | 1. After liking/unliking comment in the userFeed, actor can close comment sheet and return to userFeed. | | | |
| **Author name:** | Akbar Zeb Khan | | | |

## 4.6 System Feature 6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | 6. UserProfile | | | |
| **Summary:** | This use case is about the userProfile | | | |
| **Actors:** | User, Admin | | | |
| **Preconditions:** | 1. User and admin needs to be logged in to the application | | | |
| **Basic course/Happy**  **path:** |  | **Actor** | **Application** |  |
| 1. Actor selects the userProfile option from the main menu |  |  |
|  | 1. The application displays the actor's profile, including personal details, profile picture,cover picture , post count, like count and activity history |  |
| **Alternative path:** | None | | | |
| **Post condition:** | After viewing the UserProfile, actor can return back to dashboard | | | |
| **Author name:** | Akbar Zeb Khan | | | |

## 4.6.1 System Feature 6.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | 6.1 Add Profile/ Cover Picture | | | |
| **Summary:** | This use case is about Add Profile/ Cover Picture in the userprofile | | | |
| **Actors:** | User | | | |
| **Preconditions:** | 1. User must be already logged in to the application. 2. User must be in Userprofile | | | |
| **Basic course/Happy**  **path:** |  | **Actor** | **Application** |  |
| 1. Actor can edit the profile and cover picture from pressing the edit and change button. |  |  |
|  | 1. The application shows the page where the user can change their profile and cover picture |  |
|  |  | 1. Actor is shown an option to choose from Gallery or to select camera directly. |  |  |
|  |  |  | 1. The application saves the image in the db and saves it on the profile and cover page |  |
|  |  |  | 1. The application then displays the image. |  |
| **Alternative path:** | 1. In step 3, if the user chooses an image with a huge storage, which is too big for the application to store, it will give an error message to take another picture or to choose another image which meets the uploading requirement. | | | |
| **Post condition:** | 1. After the image has been uploaded, the application displays image | | | |
| **Author name:** | Akbar Zeb Khan | | | |

## 4.6.2 System Feature 6.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | 6.2 Edit User Posts | | | |
| **Summary:** | This use case is about editing user’s posts in userprofile | | | |
| **Actors:** | User, Admin | | | |
| **Preconditions:** | 1. User and admin needs to be logged in to the application | | | |
| **Basic course/Happy**  **path:** |  | **Actor** | **Application** |  |
| 1. Actor selects the userProfile option from the main menu |  |  |
|  | 1. The application displays the actor's profile, page |  |
|  |  | 1. Actor selects the post on the userprofile and selects the edit option. |  |  |
|  |  |  | 1. Application shows a modal to the user to the edit post. |  |
|  |  | 1. Actor edits the post. |  |  |
|  |  |  | 1. The application then saves the changes in the db |  |
| **Alternative path:** | 1. If the actor leaves the post as empty, the application will show an error and will inform the use to complete the required words count. | | | |
| **Post condition:** | 1. After viewing in the UserProfile, the post has been edited successfully. | | | |
| **Author name:** | Akbar Zeb Khan | | | |

## 4.6.3 System Feature 6.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | 6.3 Delete User Posts | | | |
| **Summary:** | This use case is about the userProfile | | | |
| **Actors:** | User, Admin | | | |
| **Preconditions:** | 1. User and admin needs to be logged in to the application | | | |
| **Basic course/Happy**  **path:** |  | **Actor** | **Application** |  |
| 1. Actor selects the userProfile option from the main menu |  |  |
|  | 1. The application displays the actor's profile page |  |
|  |  | 1. Actor taps on user post |  |  |
|  |  |  | 1. application shows a modal. |  |
|  |  | 1. Actor selects the delete button, present in the post and deletes a post. |  |  |
|  |  |  | 1. The application deletes a the post and saves the changes in the db. |  |
| **Alternative path:** | None | | | |
| **Post condition:** | 1. After viewing the UserProfile, actor can return back to userprofile | | | |
| **Author name:** | Akbar Zeb Khan | | | |

## 4.6.4 System Feature 6.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | 6.4 View User Liked Posts | | | |
| **Summary:** | This use case is about viewing user liked posts UserProfile | | | |
| **Actors:** | User | | | |
| **Preconditions:** | 1. User needs to be logged in to the application | | | |
| **Basic course/Happy**  **path:** |  | **Actor** | **Application** |  |
| 1. Actor selects the userProfile option from the main menu |  |  |
|  | 1. The application displays the actor's profile page |  |
|  |  | 1. Actor views liked posts |  |  |
|  |  |  | 1. Application displays actor’s liked posts. |  |
| **Alternative path:** | None | | | |
| **Post condition:** | 1. After viewing the UserProfile, actor can return back to userprofile | | | |
| **Author name:** | Akbar Zeb Khan | | | |

|  |  |  |
| --- | --- | --- |
| **Name of Use Case:** | 7. Edit Profile | |
| **Summary:** | This use case is about Edit Profile | |
| **Actors:** | User | |
| **Preconditions:** | 1. User must be already logged in to the application | |
|  | **Actor** | **Application** |
| **Basic course/Happy**  **path:** | 1. Actor selects the userProfile option from the main menu |  |
|  | 1. The application displays the actor's profile, including personal details, profile picture,cover picture , post count, like count and activity history |
| 1. Actor presses the Edit Profile button. |  |
|  | 5. The application shows the actor a series of options that are editable which are: First Name, Last Name, Date of Birth, Address, Email address, and Phone number and Profile Picture |
| 6. The actor presses on the section to make changes and saves them after the changes have been made |  |
|  | 7. The application will save the changes in the db and then displays it to the user on the page. |
| **Alternative path:** | 1. In step 6, if the actor makes changes that do not match the criteria of the application, the application will show an error message making the actor to type in the correct details. | |
| **Post condition:** | 1. Upon successful call the actor will be informed about the changes are successful and the page will divert back to profile page. | |
| **Author name:** | Akbar Zeb Khan | |

## 4.7 System Feature 7

## 4.8 System Feature 8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | 8. Talash-e-Gumshuda | | | |
| **Summary:** | This use case is about the missing pupil in the city | | | |
| **Actors:** | User | | | |
| **Preconditions:** | 1. User must be already logged in to the application. | | | |
| **Basic course/Happy**  **path:** |  | **Actor** | **Application** |  |
| 1. Actor selects Talash-e-Gumshuda option the from dashboard |  |  |
|  | 1. The application presents options for searching and viewing information about the missing persons. |  |
| 1. Actor searches for missing persons based on various criteria, such as name, age, gender, and location |  |  |
|  | 1. The application displays a list of missing persons with their details, including pictures and location in color-coded map, when available. |  |
|  | 1. The application provides tags for each missing person, indicating whether they have been found or are still missing. |  |
| **Alternative path:** | 1. In step 3, if there are no missing persons in the particular area then application will not display anything.. 2. In step 5, if the person is still missing actor can add more information with relevant evidence through a note/comment | | | |
| **Post condition:** | 1. After viewing the list, actor can return back to dashboard | | | |
| **Author name:** | Akbar Zeb Khan | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | 9. Color-Coded Map | | | |
| **Summary:** | This use case is about Color-coded Map | | | |
| **Actors:** | User | | | |
| **Preconditions:** | 1. User must be already logged in to the application. 2. GPS location must be on. | | | |
| **Basic course/Happy**  **path:** |  | **Actor** | **System** |  |
| 1. Actor selects the Color-Coded Map from the dashboard page. |  |  |
|  | 1. .The application displays a map showing the locations of reported incidents (actor’s current area) by a blip and different colors denoting the severity of crime per area (e.g., red for high severity, yellow for medium, green for low). |  |
|  | 1. The application displays a guide on the right side of map explaining the blip markers and the color-coding system for incident severity and weightage. |  |
| 1. Actor taps on a blip marker of a an incident on color-coded map |  |  |
|  | 1. The application displays a callout with a summary of the incident. |  |
| **Alternative path:** | 1. In step 3, Actor can zoom in and out, pan, and interact with the map to explore incidents in different locations. | | | |
| **Post condition:** | 1. After viewing the Color-coded map, actor can return back to dashboard | | | |
| **Author name:** | Akbar Zeb Khan | | | |

## 4.9 System Feature 9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | 10. Safety Analysis | | | |
| **Summary:** | This use case is about Safety Analysis | | | |
| **Actors:** | User | | | |
| **Preconditions:** | 1. User must be already logged in to the application. | | | |
| **Basic course/Happy**  **path:** |  | **Actor** | **Application** |  |
| 1. Actor selects the Safety Analysis option from the main menu. |  |  |
|  | 1. The application will request the database for the details of the crime related reports, kidnappings and other safety related report. |  |
|  | 1. The application displays safety-related data and trends, including areas with higher incident rates, types of incidents for e.g crime etc, and their frequency. |  |
|  | 1. The application displays the location, area and details of the crime with respect to the weight-age of the crime, e.g red meaning high crime, yellow moderate and green normal. |  |
| 1. The actor can categorize data by location, frequency of crime and date. |  |  |
| **Alternative path:** | 1. In step 5, if the actor has not given access to their GPS then they will not be able to view the crime of their current area. | | | |
| **Post condition:** | 1. opening the page the actor can view crime with respect to their area, with color-coded markers on the maps. | | | |
| **Author name:** | Akbar Zeb Khan | | | |

## 4.10 System Feature 10

## 4.11 System Feature 11

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | 11. Real-time Notifications | | | |
| **Summary:** | This use case is about Real-time Notification | | | |
| **Actors:** | User | | | |
| **Preconditions:** | 1. User must be already logged in to the application. 2. User must have notification enabled. | | | |
| **Basic course/Happy**  **path:** |  | **Actor** | **Application** |  |
|  | 1. Various triggers, such as the submission of a new incident report or a safety alert, prompt a real-time notification. |  |
|  | 1. The application sends push notifications to actor’s device, alerting them to the relevant post in their own area/vicinity. |  |
| 1. The actor tap on the notification |  |  |
|  |  |  | 1. The application displays the post in userFeed along with the additional details |  |
| **Alternative path:** | 1. In step 1, if the application will not receive any incoming alert, then it will not send any real-time alert to the actor. 2. In step 4, the actor can also vote for the notification’s authenticity, confirming that the incident occurred. | | | |
| **Post condition:** | 1. Upon successfully receiving notifications, the notification will be saved in alert button for the actor. | | | |
| **Author name:** | Akbar Zeb Khan | | | |

|  |  |  |
| --- | --- | --- |
| **Name of Use Case:** | 12. SOS Alert | |
| **Summary:** | This use case is about SOS alert. | |
| **Actors:** | User | |
| **Preconditions:** | 1. User must be already logged in to the application | |
|  | **Actor** | **Application** |
| **Basic course/Happy**  **path:** | 1. Actor will press the SOS Alert option in bottomBar |  |
|  | 1. The Application presents options for sending specific types of SOS alerts, such as police, ambulance, and fire brigade. |
| 1. Actor can send SOS alerts for various emergency types, including general help, ambulance, and fire brigade |  |
| 4.The actor chooses from any of the three options |  |
|  | 5. The map shows markers of nearby Police stations, Fire-Brigade stations and hospitals depending on the option chosen |
| 6. The actor presses on any marker call icon callout. |  |
|  | 7. The application will direct the screen to a phone number pop-up of the nearest particular department selected by the actor. |
| 8. The actor makes an SOS alert through phone call. |  |
|  | 9. The application will then automatically make a call to the desired extension. |
| **Alternative path:** | 1. In step 1, if the actor does not have cellular services then the SOS alert won’t be generated and an error will be shown. 2. Location services should be enabled by the actor. | |
| **Post condition:** | 1. Upon successful call the actor will inform the desired extension about their emergency | |
| **Author name:** | Akbar Zeb Khan | |

## 4.12 System Feature 12

**4.13 System Feature 13**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | 13. logout | | | |
| **Summary:** | This use case is about the logging out of the application | | | |
| **Actors:** | User, Admin and LEO | | | |
| **Preconditions:** | 1. Admin, User and LEO must be already logged in to the application. | | | |
| **Basic course/Happy**  **path:** |  | **Actor** | **Application** |  |
| 1. Actor selects the logout option from the main menu |  |  |
|  | 1. The application will send a prompt for the confirmation of log out. |  |
| 1. The actor taps the yes option |  |  |
|  |  |  | 1. The application logs the user out, ending the current session |  |
| **Alternative path:** | 1. In step 3, if the actor selects no option then the actor will remain logged in | | | |
| **Post condition:** | 1. Upon successful log out, application will display login screen. | | | |
| **Author name:** | Akbar Zeb Khan | | | |

# 

# Other Nonfunctional Requirements

## 5.1 Performance Requirements

The proposed application is developed as a Public Safety and citizen journalism system to be used by citizens of Karachi and foreigners living here along with public safety organizations. The expectations for the Crime-Safety to work flawlessly are present. Therefore

* The performance of the application will be efficient and without unnecessary delay
* User interface response in normal conditions will be instant with a maximum acceptable delay of 2 seconds
* User-posted incidents, reports, or SOS alerts will be sent to the server within 3 seconds of user submission.
* If for some reason the system crashes the data will be backed up to prevent data loss
* If there are many users using application at the same time. The system will be able to handle it without crashing or downtime.

## 5.1 Safety Requirements

* The application will adhere to strict data privacy regulations and ensure that user data is securely stored and transmitted to protect user privacy
* Real-time alert voting for each post will be enabled for users in the vicinity of incident to verify the claim.

## 5.3 Security Requirements

* To ensure security, each user will also have to provide their phone number to register, this mechanism will be used to verify identity of users during registration and login.
* User-generated content, including incident reports and user profiles, will be securely stored and transmitted to protect user data privacy.
* To prevent unauthorized access, secure database will be used. 
* Users can only login to their own profile and cannot make changes to the application system

## Software Quality Attributes

* The application will be easy to use for an average smartphone user
* The application will be responsive for various screen sizes for optimal user experience
* The application will be available to the user 24/7, with a target uptime of 99.9% to ensure that users can access it when needed.
* Crime-Safety aims to provide better usability, making it easier for users to access and utilize the app's features.

## 

## 5.5 Business Rules

* Users with the role of "Registered User" can report incidents, view the user feed, missing people list and color-coded map, and send SOS alerts
* Only Admin or LEO roles can review and remove reported content or users as needed to maintain platform integrity
* A single User can only register with 1 phone number that cannot be re-registered
* Users found indulging in posting inappropriate content will be banned

# Other Requirements

Appendix A: Glossary

* SRS : Software Requirement Specification
* SDS : Software Design Specification
* STS : Software Testing Specification
* LEO : Law Enforcement Official
* SOS : Save Our Souls
* UI : User Interface
* API : Application Programming Interface
* HTTPS : Hypertext Transfer Protocol Secure

**Software Design Specification - Crime-Safety**

This document entails all design constraints and specifications used in the project and diagrams and their usage to explain the end user all workflow and processes of Crime-Safety.

1. **Introduction**
   1. **Purpose of this document**

The aim of this document is to outline the elaborate architecture, databases and Libraries in our Application Crime-Safety. This document is designed to be any Individual that stands to be involved in or is seeking an architectural

view of the whole system. This software design specification describes the

Interface, database schemas and key diagramatic forms of the associated interfaces.Reason for selecting the specific diagrams and designs. Thereafter, this document should be read through by an an individual who has a technical background and is also not able to comprehend the interface designs provided. Additionally, a comprehensive description of the system architecture and this document will provide project elements in order to have the clarity of those things as well. The purpose of this entire document is that anyone who reads it It is possible to understand the essence using components, architecture, and designs.

* 1. **Scope of the development project**

The vision for the application is to create a platform for citizen journalism and community engagement, enabling users to share real-time updates with their fellow citizens and foster a social network focused on the safety and well-being of the people of Karachi. This all-in-one solution addresses a wide range of issues faced by Karachi's residents, including crime, road blockages, riots, protests, extreme weather conditions like heavy rains, natural disasters, or acts of terrorism. Users can actively contribute by posting reports in real-time, participating in community interactions, accessing a color-coded map for situational awareness, generating SOS alerts for urgent situations, and raising awareness about missing persons. This application aims to empower citizens with the tools and information needed to respond effectively to challenges and promote the safety and prosperity of the Karachi community.

* 1. **Definitions, acronyms, and abbreviations**
* SDS Software Design Specification
* SRS Software Requirement Specification
* STP Software Test Plan
* NFR Non-Functional Requirements
* IEEE Institute of Electrical and Electronics Engineers
* UX User Experience
* UI User Interface
* MySQL My Structured Query Language
* HTTPS Hypertext Transfer Protocol
* LEO : Law Enforcement Official
* SOS : Save Our Souls
* UI : User Interface
* API : Application Programming Interface

**1.4 References**

<https://www.researchgate.net/publication/262398198_Mobile_applications_for_incident_reporting_systems_in_urban_contexts_Lessons_learned_from_an_empirical_studys>

<https://dl.acm.org/doi/abs/10.1145/3290607.3312781>

<https://bmcpublichealth.biomedcentral.com/articles/10.1186/s12889-022-13551-9>

<https://citizen.com>

* 1. **Overview of document**

This Software Design Specification is structured such that an individual to comprehend the whole goal, tasks, and hierarchy of our Community Safety Social Media Application through this document as it progresses. Firstly, We would find the summary of this document which specifies why is it that we need to establish this document and the range of it with references where help was received from generate this document. In the second portion, we present the system architecture consisting of general constraints, data design and program structure of the application. Second, a summary of the features and interface. The design of the application is given. Finally, the architecture of the has been incorporated in application with the help of several diagrams which an ordinary person can comprehend everything the features of the system.

**2. System architecture description**

* 1. **Section Overview**

We use MySQL database to map entities for Crime-Safety Application. Database will impose data limitations and relations to represent the data in more organized way using data dictionary and normalized tables.

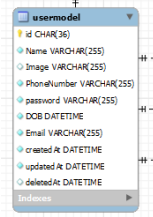
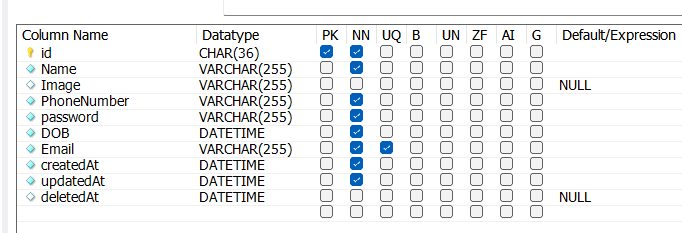
* 1. **General Constraints**

Functions of social media are posting, commenting, liking and others. Then we have user profile and cover images, This therefore increases the space needed to save what user provides. To scale up the project, we’ll require substantial cloud storage as videos take a lot of storage and as images are very bulky, so we will look to purchase Amazon EC2 and S3 for extra space.

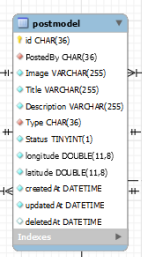
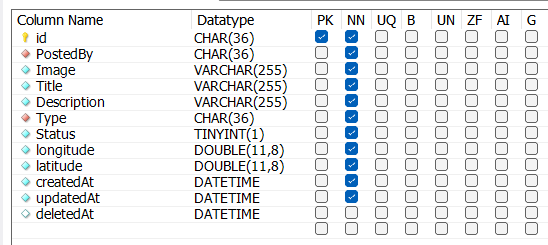
The application will be developed using:

* React Native
* Node.js
* Express.js
* MySQL
* Android Studio
* Google Map API
* REDUX
* REST API
  1. **Data Design**

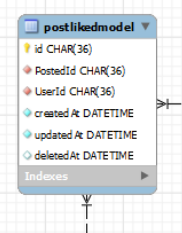
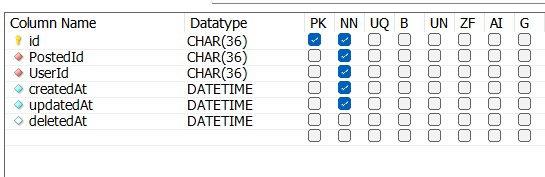
**2.3.1 USERS**



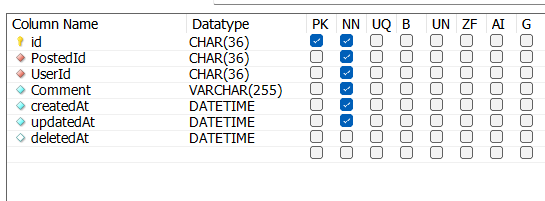
**2.3.2 POSTS**

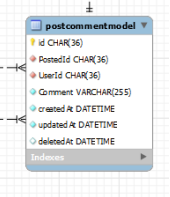


**2.3.3 POST LIKED MODEL**

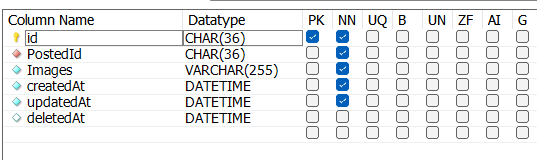


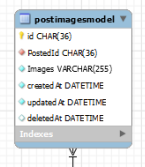
**2.3.4 POST COMMENT MODEL**



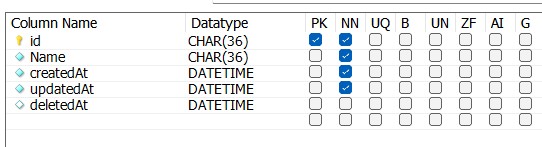


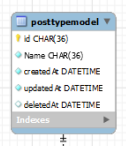
**2.3.5 POST IMAGES MODEL**



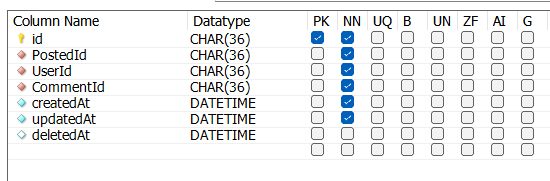


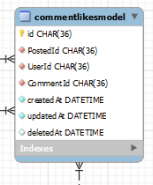
**2.3.6 POST TYPE MODEL**



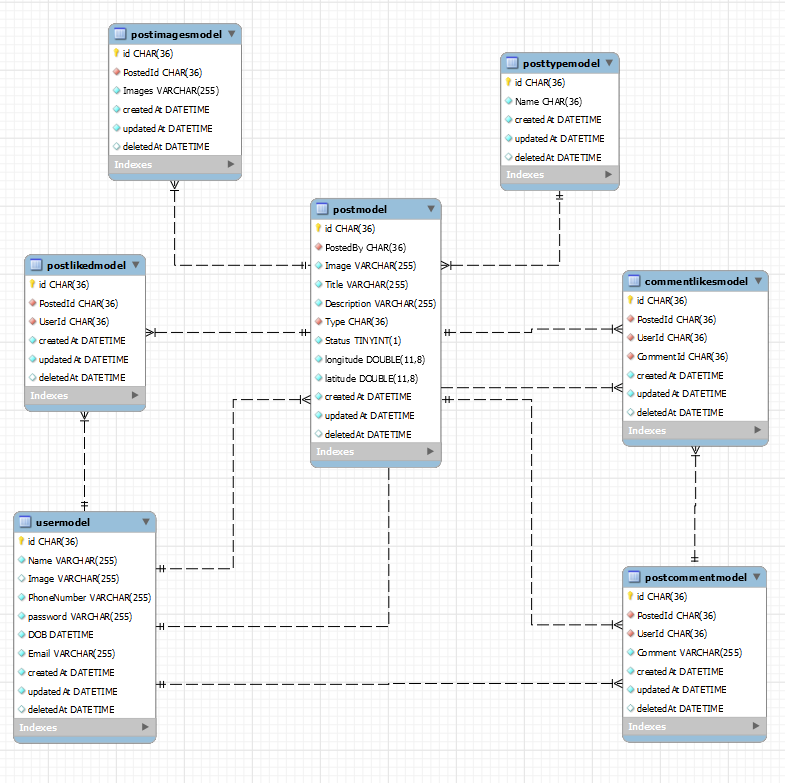


**2.3.7 Comment Likes Model**



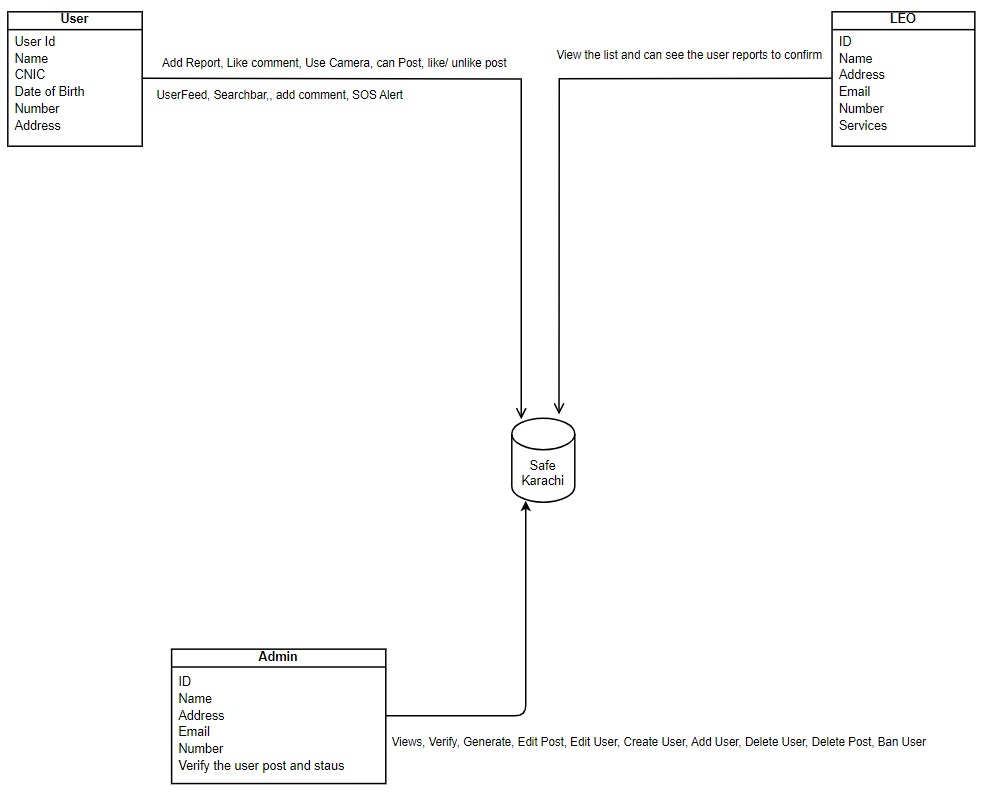


**2.3.8. Entity Relation Diagram**



* 1. **Program Structure**

**CLASS DAIGRAM:**

****

* 1. **Alternatives Considered**

No Alternatives considered

1. **Detailed description of components**  
     
   All components of our app as of yet are detailed out here  
   1. Section Overview  
      Provide a summary of the contents of this section
   2. Component n Detail (include a sub-section for each component)

* Login/Sign Up

|  |  |
| --- | --- |
| Identification | Login/User |
| Type | Class/Form/Method |
| Purpose | The login/Signup screen authenticates and allow user, admin or LEO to login or create the account. |
| Function | This screen navigates to our dashboard page, user inputs email and password that is validated from our database. |
| Subordinates | User Authentication, Login form |
| Dependencies | Formik, Axios, React Native Paper |
| Interfaces | The screens are designed to give credentials to the overall users of the system which provides the services for further priorities and procedure. |
| Resources | * Software Requirements: MySql Database, Expo. * Database Requirements: It validates overall users that exists in the database. |
| Processing | The screens require processing when user inputs email and password or to create an account to access the application. |
| Data | Once user creates the account all the relevant fields (Name, Email,Password,DOB, Phone number and Password) are stored in  the database. |

* Add Report

|  |  |
| --- | --- |
| Identification | Add report |
| Type | Class/Form/Method |
| Purpose | Allows users to submit a report detailing an incident or safety concern. |
| Function | Navigates to a form where users input information about an incident, which is then submitted to the database and system. |
| Subordinates | None |
| Dependencies | Formik, Axios, React Native Paper, React Native Maps |
| Interfaces | Provides a user-friendly form for users to input details about an incident or safety concern.  . |
| Resources | * Software Requirements: MySql Database, Expo. * Database Requirements: Stores submitted reports in the database. |
| Processing | Involves processing user-inputted data, validating and storing it in the database. |
| Data | Captures and stores information such as incident details, location, image/video, and user information in the system's database. |

* Camera

|  |  |
| --- | --- |
| Identification | Camera Feature |
| Type | Module/Feature |
| Purpose | Allows users to capture photos or videos using the device camera, toggle between front and back cameras, and utilize the flashlight. |
| Function | Integrates the Expo Camera API to provide functionalities for taking pictures, recording videos, switching between front and back cameras, and controlling the flashlight. |
| Subordinates | None |
| Dependencies | Expo Camera, Expo Image Picker, Device Camera Hardware |
| Interfaces | * Camera View for capturing photos/videos * Flashlight toggle * Front/Back camera switch * Integration with gallery for image selection. |
| Resources | Software Requirements: Expo Camera, Expo Image Picker  Hardware Requirements: Device with a camera and flashlight |
| Processing | Utilizes the device camera APIs for capturing media.  Integrates with the device's flashlight control. |
| Data | Captures and stores photos or videos in the device's gallery or another shows in add report bottomSheet. |

* UserFeed

|  |  |
| --- | --- |
| Identification | UserFeed |
| Type | Module/Feature/Class |
| Purpose | Displays a personalized feed of content for the user. |
| Function | Gathers and displays posts from database |
| Subordinates | Like/Unlike Post, Like/Unlike comment, Add Comment |
| Dependencies | Axios, Moment.js, Reatc Native Paper |
| Interfaces | User interface displaying the personalized feed, Interaction options (like, comment, share), Content retrieval mechanisms. |
| Resources | Software Requirements: MySql Database |
| Processing | Utilizes the database to display user posts |
| Data | Displays content such as posts, articles, or updates tailored to the user's interests and interactions. |

* Like/Unlike Post/Comment

|  |  |
| --- | --- |
| Identification | Like/Unlike Feature |
| Type | Module/Feature |
| Purpose | Allows users to express approval or disapproval of posts or comments. |
| Function | Provides users with the ability to like or unlike posts and comments, influencing engagement and feedback. |
| Subordinates | None |
| Dependencies | Axios,MySql database, user authentication |
| Interfaces | Like and unlike buttons associated with posts and comments, User feedback counters. |
| Resources | Software Requirements: MySql Database, counter to record likes |
| Processing | Registers user interactions when liking or unliking posts or comments, updating relevant counters. |
| Data | Records user likes, maintains counts of likes and unlikes for posts and comments. |

* SearchBar

|  |  |
| --- | --- |
| Identification | SearchBar |
| Type | Module/Feature |
| Purpose | Facilitates users in searching for specific content within the userFeed |
| Function | Allows users to search for a post by postType, post tile and post description |
| Subordinates | Search API |
| Dependencies | Axios,MySql database, user authentication |
| Interfaces | Comment input field, Submit button, Comment display section. |
| Resources | Software Requirements: MySql Database, user search input |
| Processing | Takes user input, processes search query, retrieves relevant data from the database, and displays results. |
| Data | User search queries, Search results. |

* UserProfile

|  |  |
| --- | --- |
| Identification | UserProfile |
| Type | Module/Feature/Class |
| Purpose | Displays and manages user profile information, including profile/cover images, name, bio, follower count, posts liked count, posts posted count, and a list of posts posted and liked. |
| Function | Allows users to view their profile |
| Subordinates | Profile image, Cover image, User data display components, Edit Profile |
| Dependencies | User data from the application database. |
| Interfaces | User profile image, Cover image, Name, Bio, Follower count, Posts liked count, Posts posted count, List of posts posted, List of posts liked.  Resources: User data, User profile image, Cover image, Posts data. |
| Resources | Software Requirements: MySql Database, user model database |
| Processing | Retrieves user data from the application database, organizes and formats it for display. |
| Data | User profile image, Cover image, User name, Bio, Follower count, Posts liked count, Posts posted count, List of posts posted, List of posts liked. |

* Edit User Post

|  |  |
| --- | --- |
| Identification | Edit User Post |
| Type | Feature |
| Purpose | Allows users to edit the content of their own posts displayed in the user profile. |
| Function | Allows users to edit their posts that they have posted |
| Subordinates | None |
| Dependencies | User post data from the application database. |
| Interfaces | * User's Posts Section: Displays a list of posts created by the user in their UserProfile. * Edit Post UI Component: Activated when the user chooses to edit a specific post. It provides fields and options to modify the post content. |
| Resources | * User Data: Information about the user and their posts. * Post Data: Content and details of the posts created by the user. |
| Processing | * User Initiated Edit Action: Triggered when the user selects the option to edit a specific post. * Edit Post Component Activation: The Edit Post UI component is activated, allowing the user to modify the post. * Update in Database: When the user finalizes the changes, the updated post information is saved back to the application database. |
| Data | * Original Post Content: Text, media, and details of the post before editing. * Modified Post Content: Text, media, and details of the post after editing. |

* Delete User Post

|  |  |
| --- | --- |
| Identification | Delete User Post |
| Type | Feature |
| Purpose | Allows users to delete their own posts from the UserProfile or feed. |
| Function | Provides the functionality for users to permanently remove a post they have created. |
| Subordinates | None |
| Dependencies | User post data from the application database. |
| Interfaces | * User's Posts Section: Displays a list of posts created by the user in their UserProfile. * Delete Post Option: Presented when the user selects a specific post, triggering the deletion process. * Confirmation Dialog: A dialog confirming the user's intent to delete the post, preventing accidental deletions. |
| Resources | * User Data: Information about the user and their posts. * Post Data: Content and details of the posts created by the user. |
| Processing | * User Initiated Delete Action: Triggered when the user selects the option to delete a specific post. * Confirmation Dialog: Ensures that the user is certain about deleting the post. * Post Deletion: Removes the selected post from the user's posts list and the application database. |
| Data | * Post Content: Text, media, and details of the post to be deleted. |

* Edit Profile

|  |  |
| --- | --- |
| Identification | Edit Profile |
| Type | Feature/Module/Class |
| Purpose | Allows users to modify and update their profile information. |
| Function | Provides a user-friendly interface for users to edit and customize their profile details. |
| Subordinates | User's profile data, Input fields for editing |
| Dependencies | User post data from the application database. |
| Interfaces | * Edit Profile Option: Present in the UserProfile, accessible to the user. * Editable Fields: Name, profile picture,, email, phone Number, DOB and Password. * Save Changes Button: Enables users to confirm and save the edited information. * Cancel Button: Allows users to discard changes and revert to the previous state. |
| Resources | * User Data: Existing user information, including name, bio, profile picture, cover photo, and contact details. * Edited Data: Information entered or modified by the user during the editing process. |
| Processing | * User Initiated Edit Action: Triggered when the user selects the option to edit their profile. * Editing Interface: Displays editable fields with the current information pre-filled. * User Input: Accepts modifications to the existing information. * Validation Checks: Ensures that the entered information is valid and follows any specified constraints. * Profile Update: Saves the edited information to the user's profile. |
| Data | Edited Profile Information: Modified Name, profile picture, email, phone Number, DOB and Password. |

* SOS Alert

|  |  |
| --- | --- |
| Identification | SOS Alert |
| Type | Feature/Module |
| Purpose | Allows users to access emergency services locations nearby and generate a call for assistance. |
| Function | Provides a quick and accessible interface for users to seek emergency help, view nearby service locations, and initiate a call. |
| Subordinates | Location services, Emergency service database, Call initiation function. |
| Dependencies | react-native-phone-call, react native maps, location services |
| Interfaces | * SOS Alert Button: Present in the application, accessible to the user for quick activation. * Nearby Service Locations Map: Displays a map with markers indicating nearby emergency service locations. * Call Initiation Button: Allows the user to initiate a call to the selected emergency service provider. |
| Resources | * Location Services: Utilizes the device's GPS or other location services to determine the user's current location. * Emergency Service Database: Stores information about nearby emergency service providers, including contact details. * Calling Functionality: Utilizes the device's call features to initiate a call to the selected service. |
| Processing | * SOS Activation: Triggered when the user selects the SOS Alert button. * Location Retrieval: Determines the user's current location using device location services. * Nearby Service Query: Retrieves information about nearby emergency service providers. * Map Display: Shows a map with markers indicating the locations of nearby emergency services. * Call Initiation: Allows the user to initiate a call to the selected service provider. |
| Data | * User's Location: Coordinates (latitude, longitude) indicating the user's current location. * Emergency Service Information: Contact details of nearby emergency service providers. * Call Log: Records information about initiated calls for emergency assistance. |

* Talash-e-Gumshuda

|  |  |
| --- | --- |
| Identification | Talash-e-Gumshuda |
| Type | Feature/Module/Class |
| Purpose | Allows users to search for missing persons and report information related to missing individuals. |
| Function | Provides a platform for users to report missing persons, view details about missing individuals, and contribute information to help locate them. |
| Subordinates | Missing Persons Database, User Reporting Interface, Search and View Functions. |
| Dependencies | Database of missing persons, User input for reporting, Search algorithms. |
| Interfaces | * Report Missing Person Form: Allows users to input details about a missing person and submit a report. * Search Functionality: Enables users to search for missing persons based on various criteria. * Missing Person Details: Displays information about missing individuals reported by users. |
| Resources | * Missing Persons Database: Stores information about reported missing persons, including details provided by users. * User Reporting Interface: Allows users to submit reports with information about missing individuals. * Search Algorithms: Facilitates efficient searching for missing persons based on different parameters. |
| Processing | * Report Submission: Users input details about missing persons and submit the report. * Database Update: Information from user reports is stored in the Missing Persons Database. * Search Processing: Users can search for missing persons using different criteria. * Display Missing Person Details: Information about missing individuals is retrieved and displayed. |
| Data | * User-Reported Information: Details provided by users when reporting a missing person. * Missing Persons Database: Records containing information about missing individuals, including their descriptions, last known locations, etc. * Search Criteria: Parameters used by users to search for missing persons (e.g., name, location, etc.). |

* Safety Analysis

|  |  |
| --- | --- |
| Identification | Safety Analysis |
| Type | Feature/Module/Class |
| Purpose | Provides users with visualized safety analysis, including area-wise crime rates, temporal trends, and infographics based on the data in the system. |
| Function | Analyzes crime data from the system and presents it to users in the form of charts, graphs, and infographics for better understanding and decision-making. |
| Subordinates | Data Analysis Module, Visualization Components, User Interface. |
| Dependencies | Crime Data from the System, Data Analysis Algorithms, Visualization Libraries. |
| Interfaces | * Area-Wise Crime Rate Map: Displays a map with marked areas indicating crime rates. * Temporal Trends Chart: Graphical representation of crime trends over time. * Infographics: Visual representations providing insights into key safety metrics. * User Interface: Allows users to interact with and explore safety analysis features. |
| Resources | * Crime Data: Data from the system's crime reports and incidents. * Data Analysis Algorithms: Algorithms for processing and analyzing crime data. * Visualization Libraries: Tools for creating charts, graphs, and infographics. * User Interface Components: Elements enabling user interaction and exploration of safety analysis. |
| Processing | * Data Analysis: Utilizes algorithms to process crime data and extract meaningful insights. * Visualization Generation: Transforms analyzed data into visual representations such as charts, graphs, and infographics. * User Interaction: Allows users to interact with the visualized data, exploring different metrics and time frames. |
| Data | * Crime Data: Information about reported incidents, including location, time, and type of crime. * Analysis Results: Outputs from data analysis algorithms, including crime rates, trends, and safety metrics. * Visualization Components: Charts, graphs, and infographics generated for user consumption. |

* Real-time Notification

|  |  |
| --- | --- |
| Identification | Safety Analysis |
| Type | Feature/Module |
| Purpose | Provides users with instant notifications about incidents or reports posted in their current area. Enables user interaction through voting (like and unlike) to validate the reported events. |
| Function | Delivers real-time push notifications to users when new incidents or reports are posted, allowing them to vote on the credibility of the information. |
| Subordinates | Data Analysis Module, Visualization Components, User Interface. |
| Dependencies | Push Notification Service, User Interaction Module, Incident Reporting Module. |
| Interfaces | * Push Notification: Delivers instant notifications to users' devices. * User Voting Interface: Allows users to interact with notifications by voting (like or unlike). * Incident Reporting Integration: Coordinates with the incident reporting module for real-time updates. |
| Resources | * Incident Reports: Data about reported incidents, including location, time, and details. * User Preferences: Settings related to notification preferences and voting behavior. * Push Notification Service: External service responsible for delivering notifications to users' devices. |
| Processing | * Notification Delivery: Sends push notifications to users' devices in real-time. * User Interaction: Enables users to vote on the credibility of reported incidents through the voting interface. * Incident Reporting Integration: Collaborates with the incident reporting module to ensure real-time updates are reflected in notifications. |
| Data | * Incident Reports: Information about reported incidents, including location, time, and details. * User Votes: Data reflecting user votes (like or unlike) on specific incidents. * Notification Status: Records indicating whether a notification has been delivered and viewed by the user. |

* Logout

|  |  |
| --- | --- |
| Identification | Safety Analysis |
| Type | Feature |
| Purpose | Allows users to log out securely. |
| Function | Terminates the session, clears user data, and navigates to login. |
| Subordinates | User Authentication Module, Session Management Module. |
| Dependencies | User Account Data, User Interface Components. |
| Interfaces | * User Interface Component: Provides a user interface element (e.g., button) for initiating the logout process. * Authentication Module Integration: Coordinates with the user authentication module to ensure a secure logout process. * Session Management: Manages the termination of the user's current session. |
| Resources | * User Account Data: Information associated with the user's account, such as session tokens and preferences. * User Interface Components: Elements (e.g., buttons) in the user interface that trigger the logout process. |
| Processing | * Session Termination: Initiates the process to securely terminate the user's current session. * Data Clearance: Clears locally stored user data from the device to ensure privacy. * User Interface Update: Navigates the user to the logout state or login screen, providing feedback about the successful logout. |
| Data | * User Account Data: Information associated with the user's account, which is cleared during the logout process. * Session Status: Indicates whether the user is currently logged in or logged out. |

# User Interface Design

* 1. Section Overview

This portion offers a glimpse into the user interface of the Beauticia app. It provides a detailed explanation of the chosen user interface components and the rationale behind their selection for this product. The section briefly explores the concept and design principles guiding the user interface and its various components.

* 1. Interface Design Rules

The design rules used in our interface are as follows:

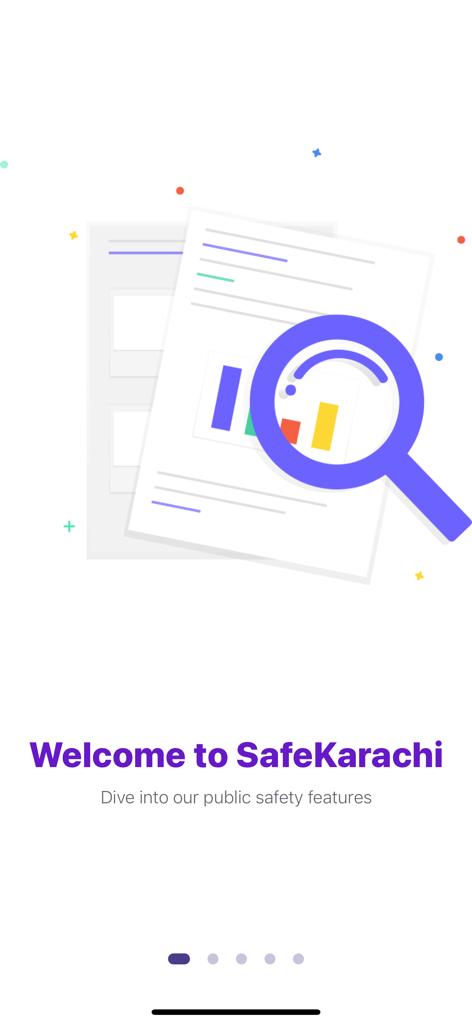
* The interface is flexible and user friendly.
* The added Buttons are easy to use.
* Standard Social Media application UI standard is used for this project.
* It allows user to directly manipulate the interface objects.
* MUI Icons are used for our design and icons  
  1. GUI Components

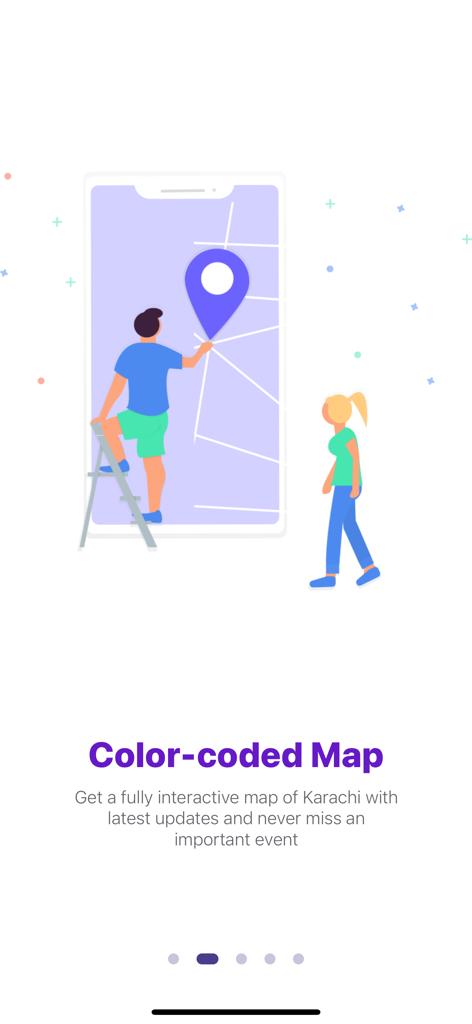
Note the GUI components or API's provided in the development environment that you plan on using.

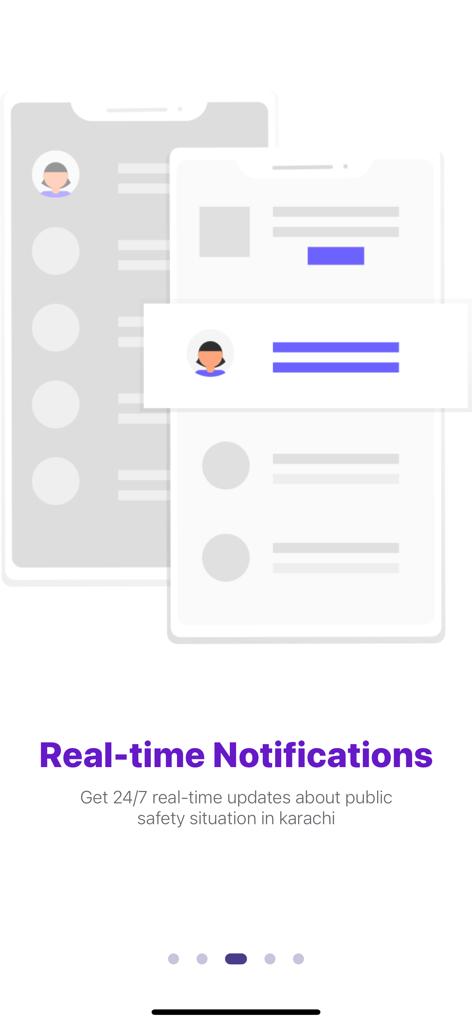
* 1. Detailed Description

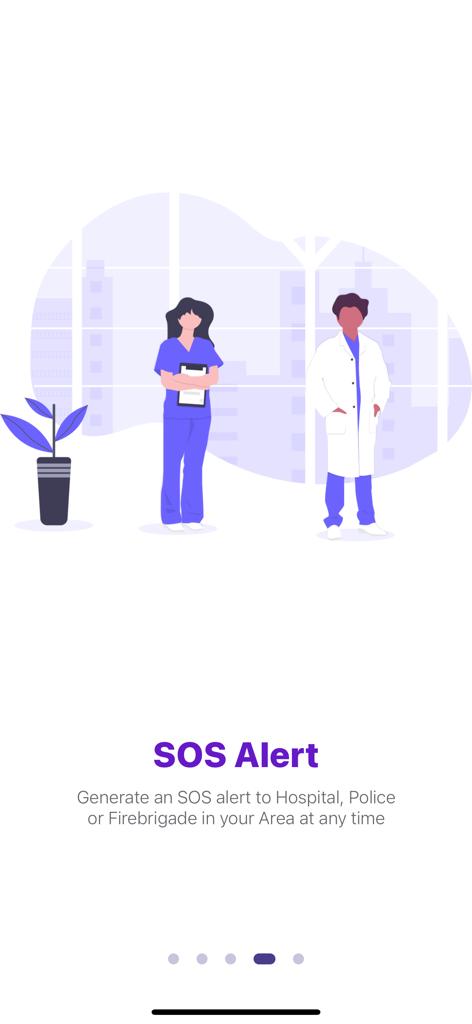
Provide a detailed description of the user interface including screen images. You may prefer to reference an appendix containing the screen snapshots.

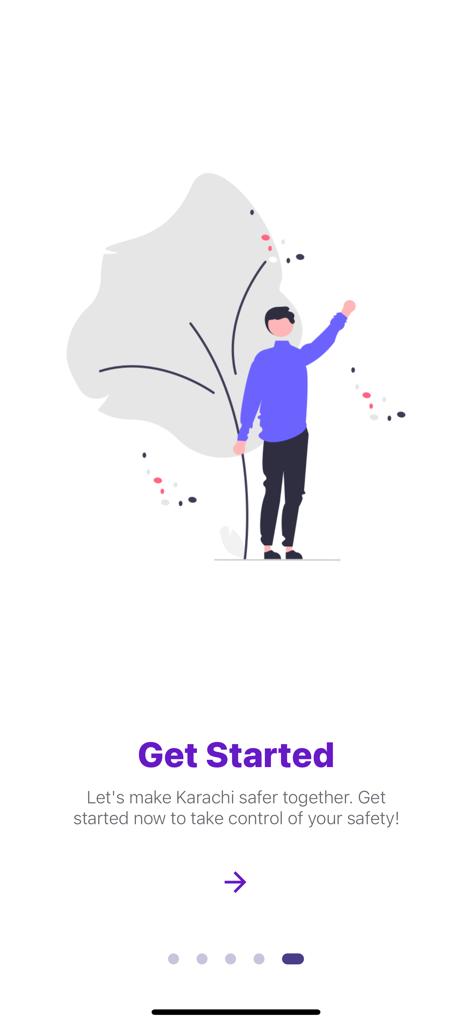
DashBoard:



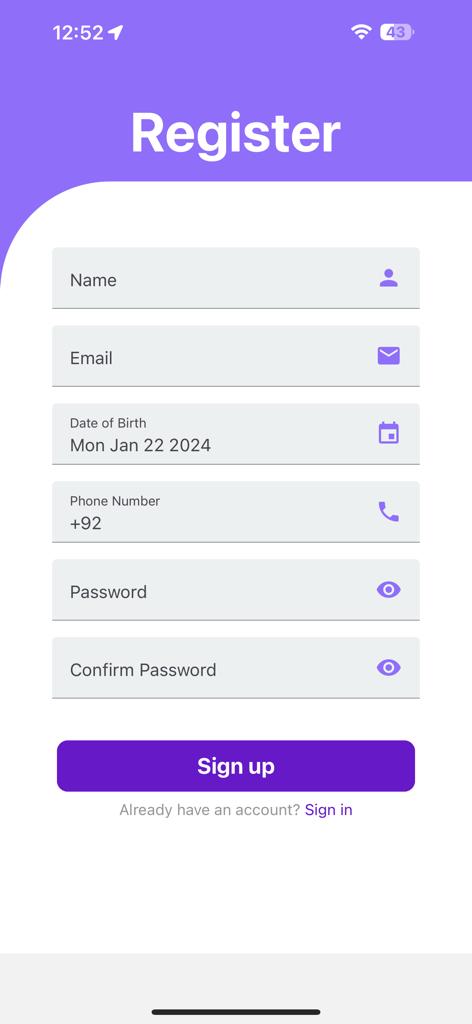




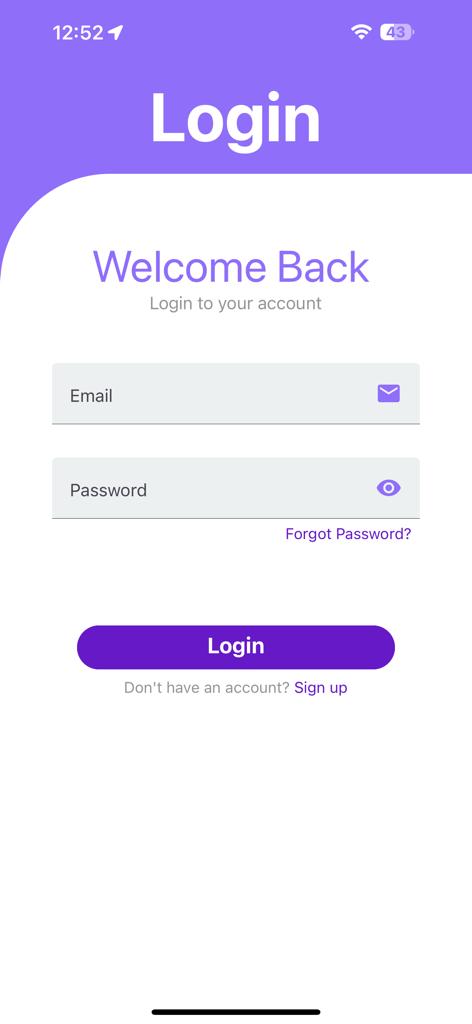




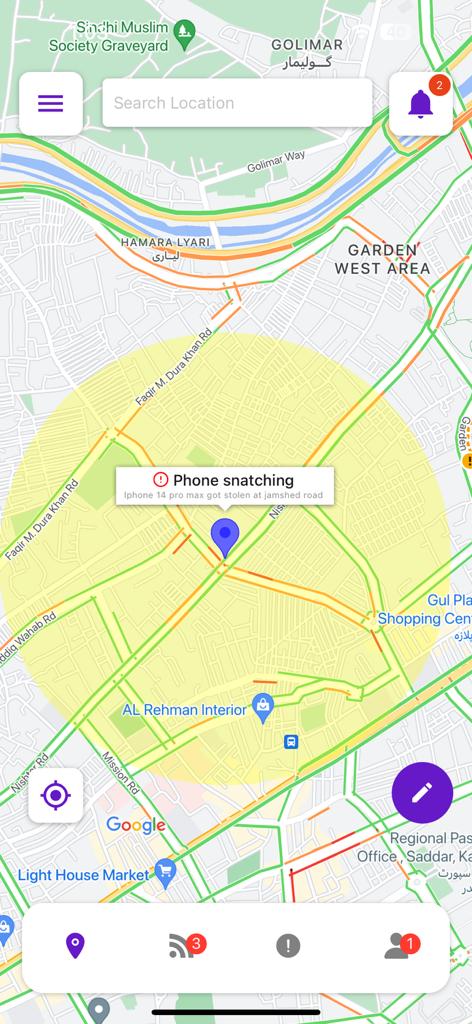
Registration/Signup:



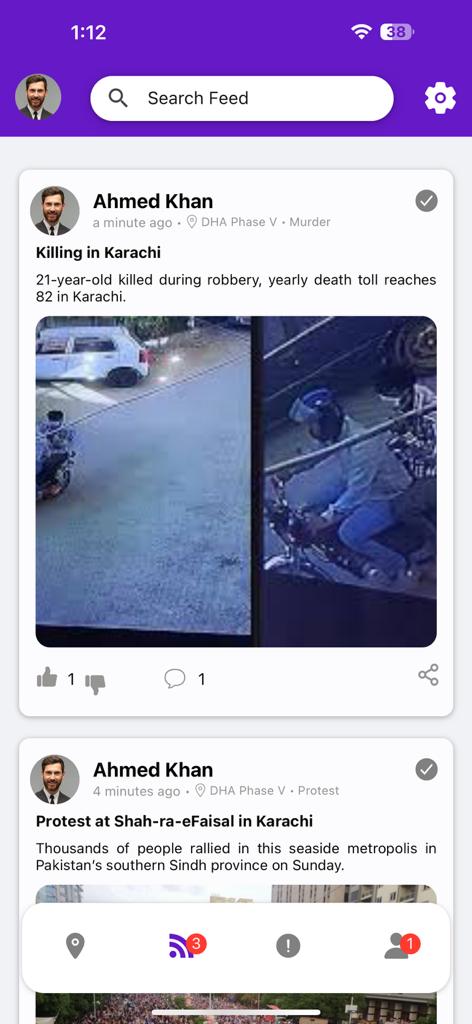
Login Page:

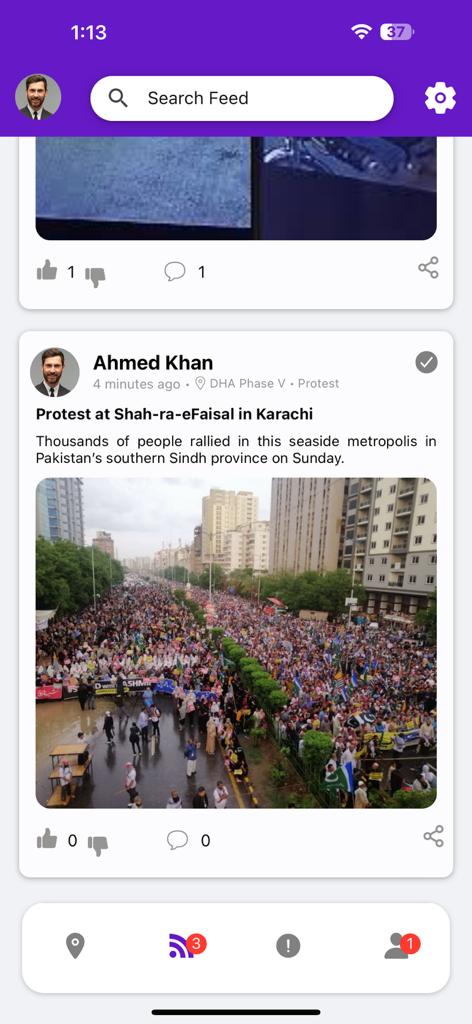


Map Page:

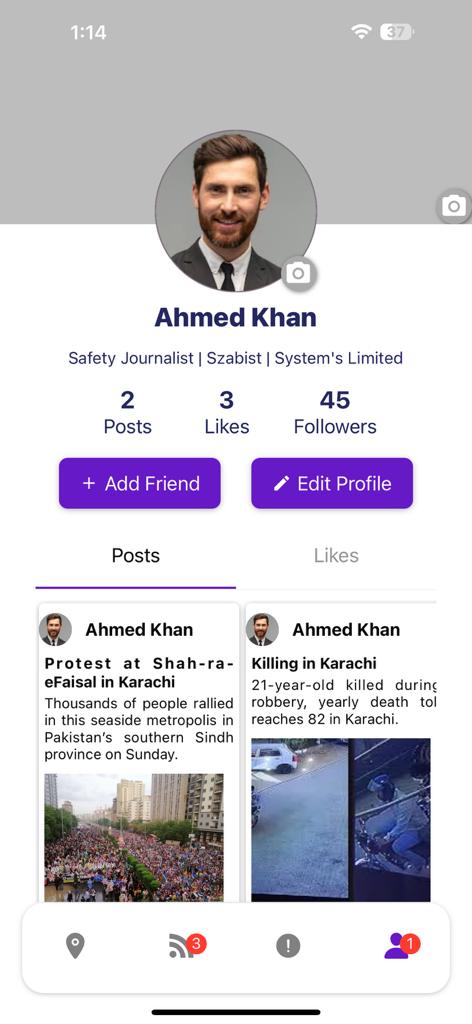


UserFeed:

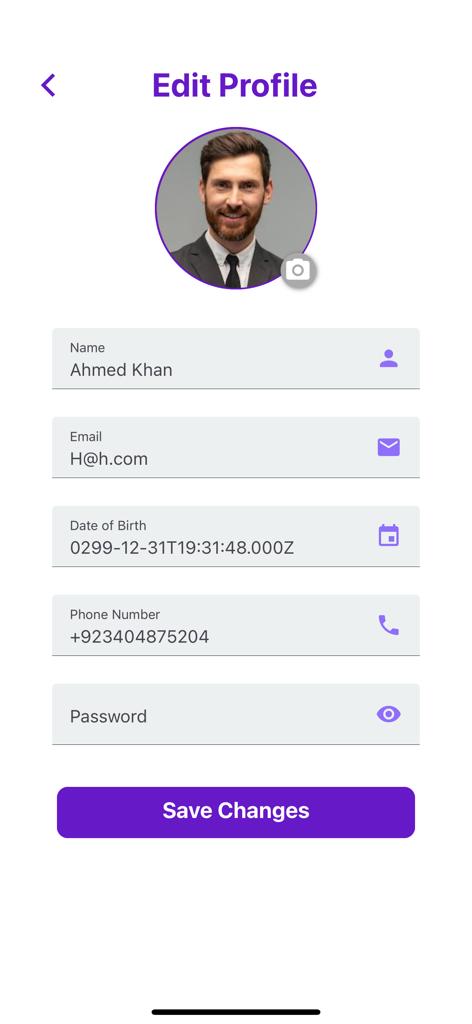




UserProfile:



Edit Profile:



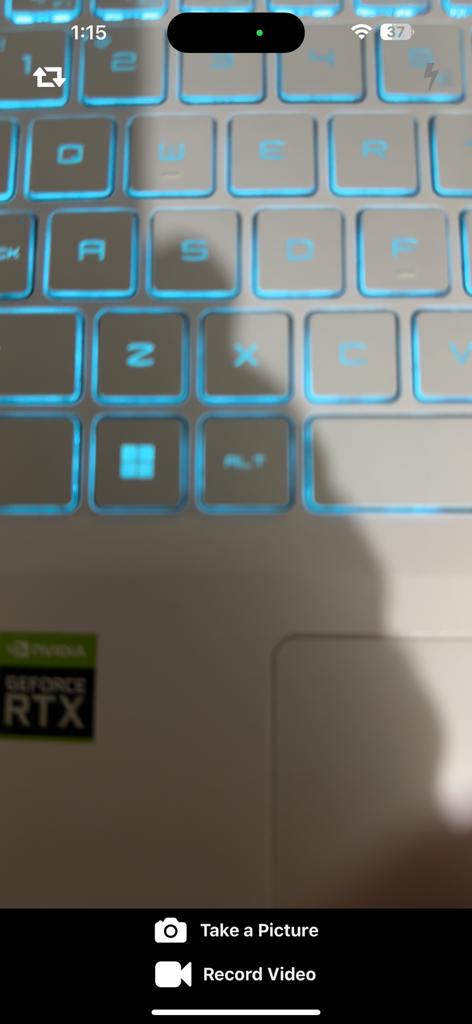
User Post(Delete Post):



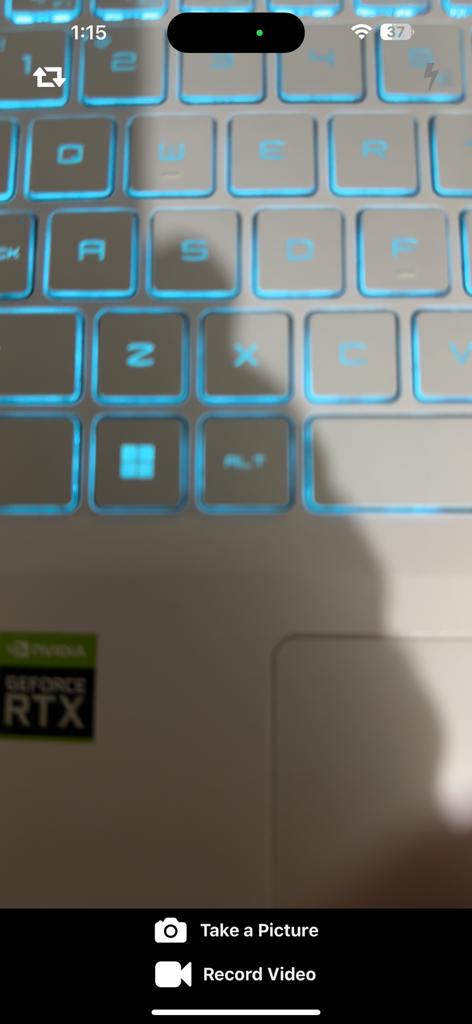
User Post(Edit Post):



Camer/Upload Picture:



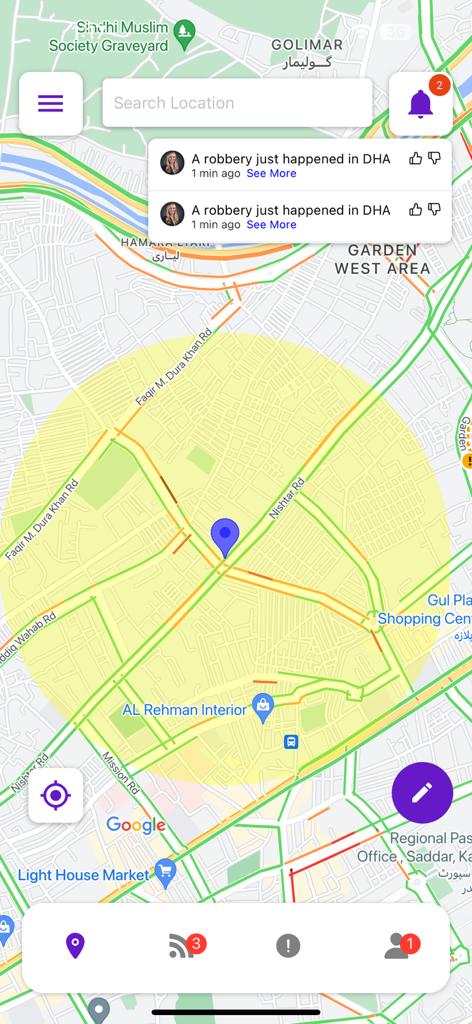
Camer/Upload Video:



Create Post:



Notification:



**5.0 Reuse and relationships to other products**

For teams doing enhancement work, reuse is an important issue. Most enhancement work should focus on extending, rather than replacing, the design and product development from earlier semesters. For teams doing new development, reuse can also be an important strategy. In some cases, there is freeware that could be incorporated. In other cases, there are existing modules or classes that could be adapted. Another possibility is the use of special tools that produce open source results and thus permissible under the terms of this course.

This section should include the following subsections as appropriate:

* how reuse is playing a role in your product design
* how reuse is playing a role in your product implementation (and the motivation for changes)
* if you are not reusing material that is available, then give motivation for why it is being thrown out.

**6.0 Design decisions and tradeoffs**

Use this section to motivate any decisions that will help the reader understand the design that your team is using. This section can also capture good ideas that were abandoned and the reasons for leaving them out of the design.

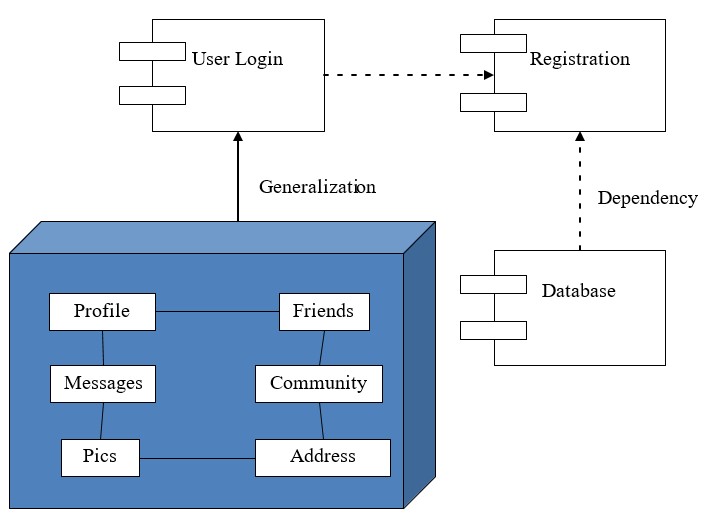
**7.0 Pseudocode for components**

**8.0 Appendices**

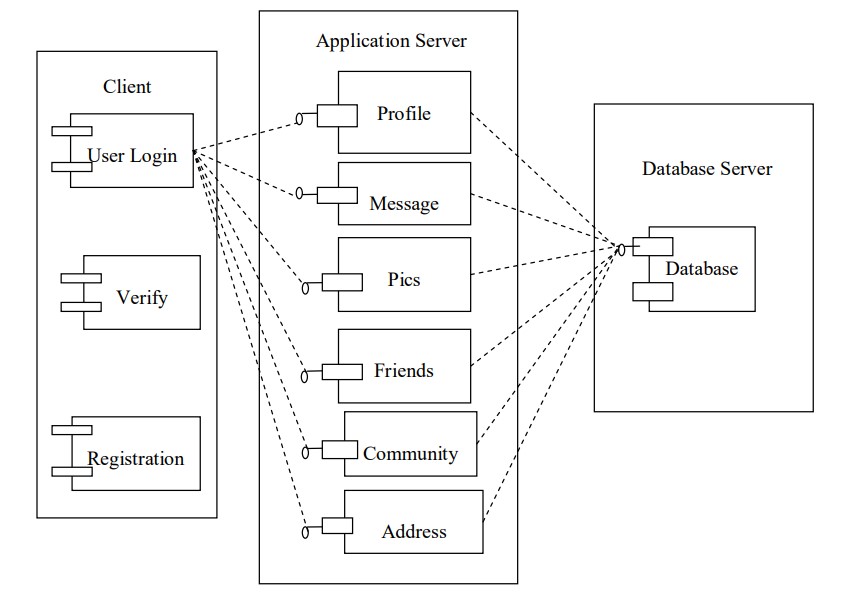
The following list presents the diagrams that should be included at appropriate places

|  |  |
| --- | --- |
| Class Diagram | Describes the structure of a system |
| Object Diagram | Expresses possible object combinations of a specific Class Diagram |
| Statechart Diagram | Expresses possible states of a class (or a system) |
| Activity Diagram | Describes activities and actions taking place in a system |
| Sequence Diagram | Shows one or several sequences of messages sent among a set of objects |
| Collaboration Diagram | Describes a complete collaboration among a set of objects |
| Use-case Diagrams | Illustrates the relationships between use cases |
| Component Diagram | A special case of a Class Diagram used to describe components within a software system |
| Deployment Diagram | A special case of a Class Diagram used to describe hardware within the overall system architecture |
| System Block diagram | A diagram showing the major components of the system with its interconnections and external interfaces |

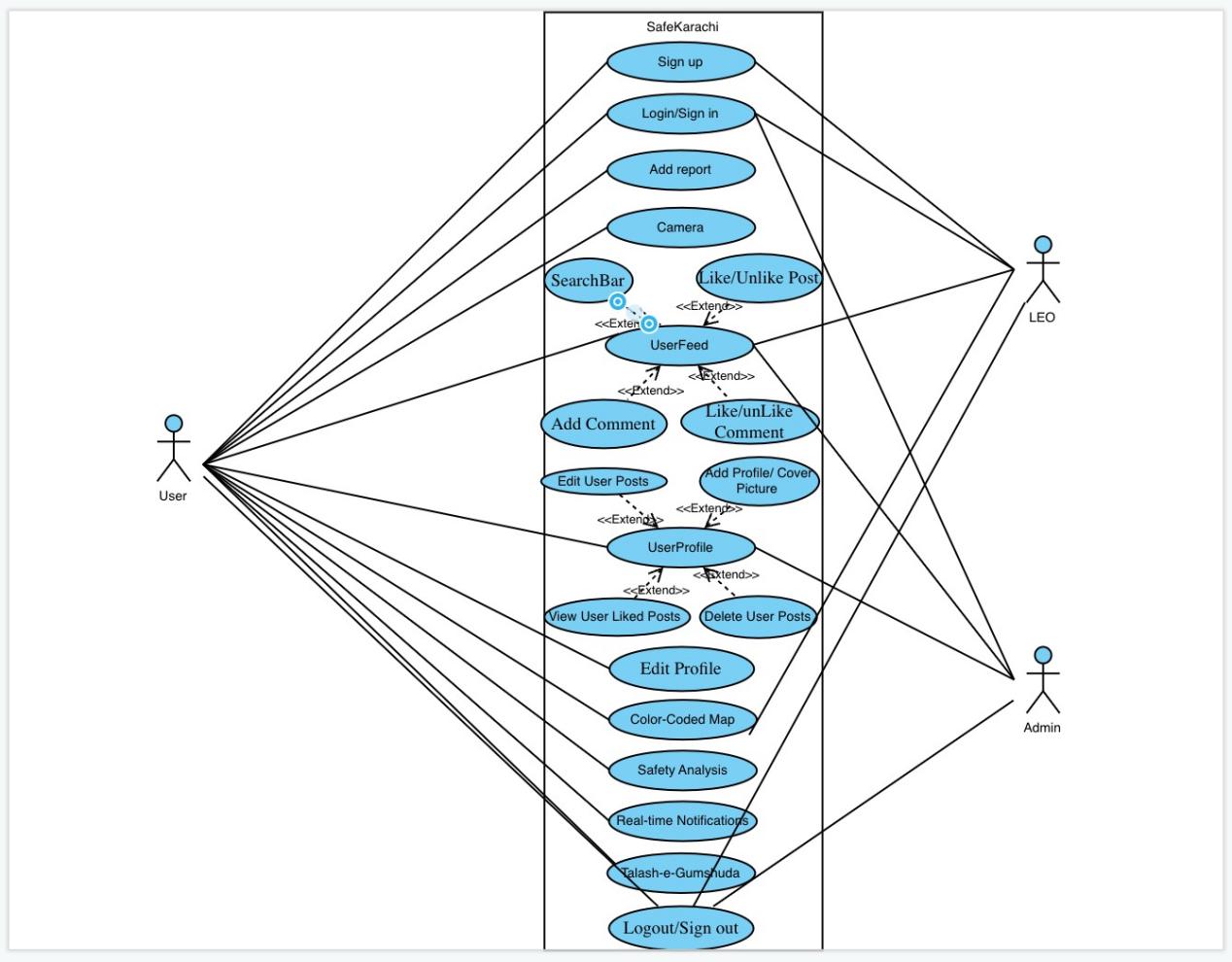
* **Component Diagram:**



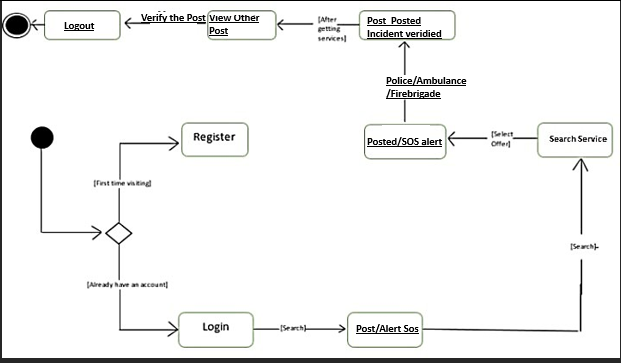
* **Deployment Diagram:**



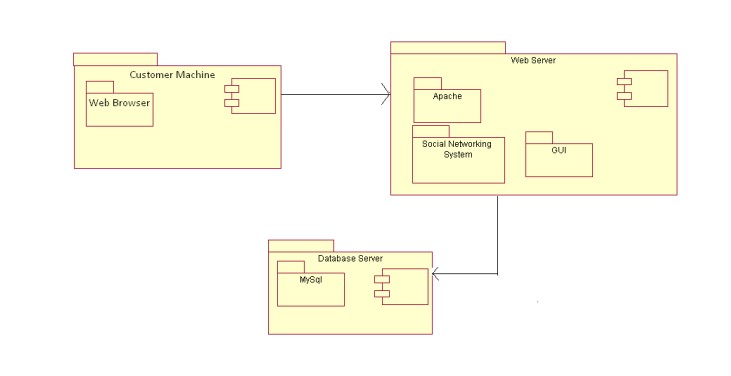
USE CASE DIAGRAM



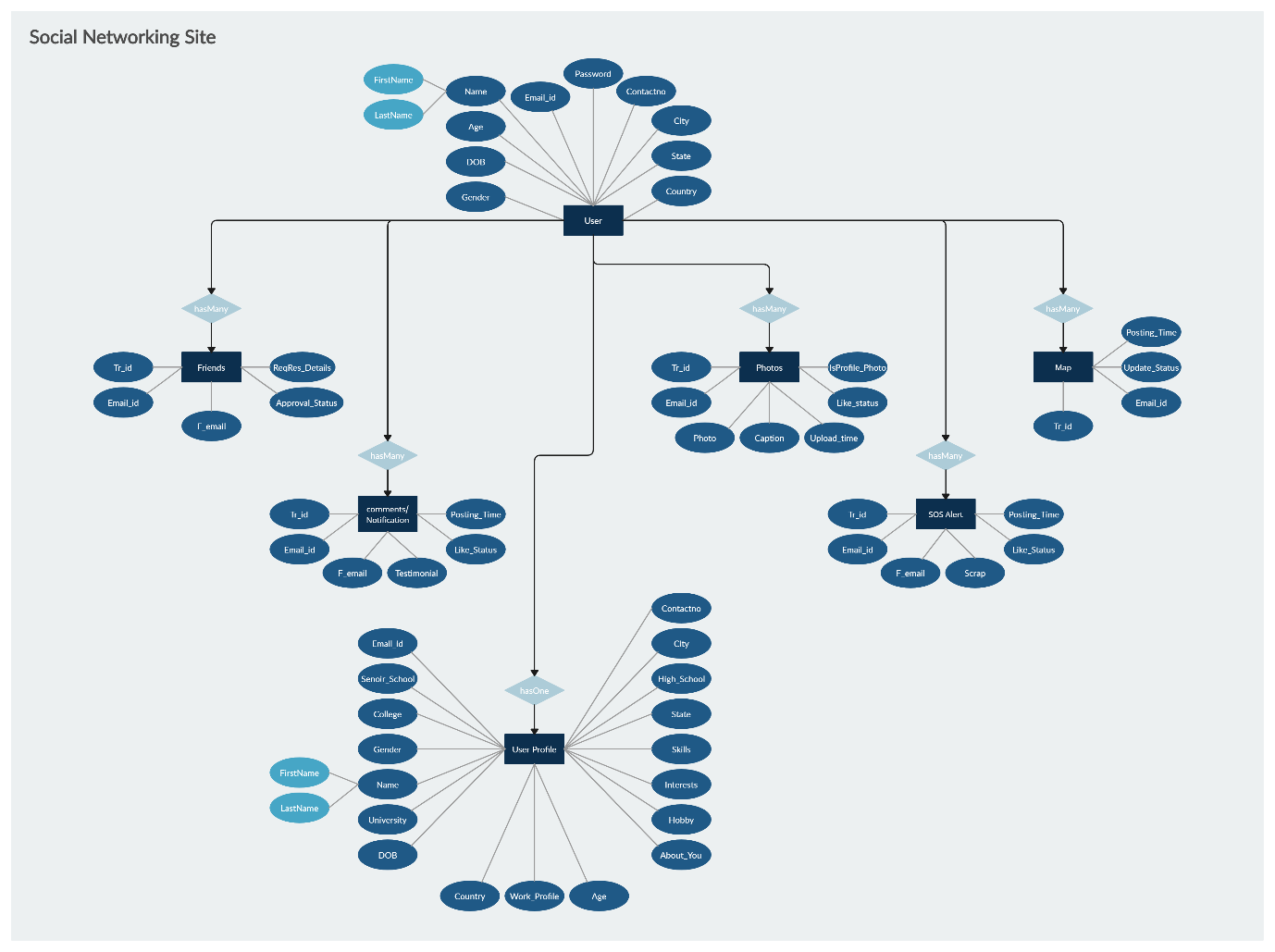
* **Statechart Diagram**



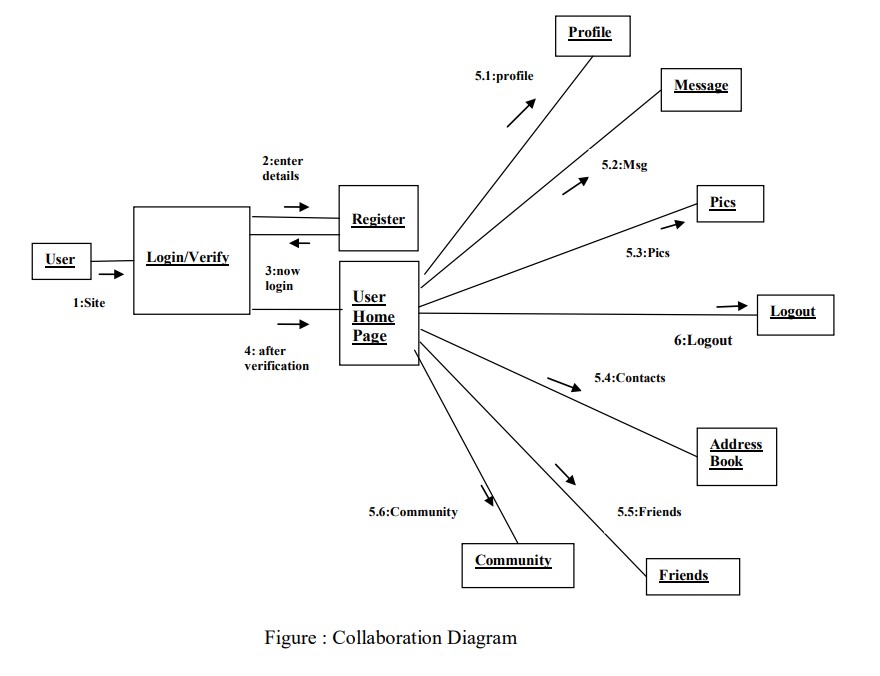
* **System Block diagram**

****

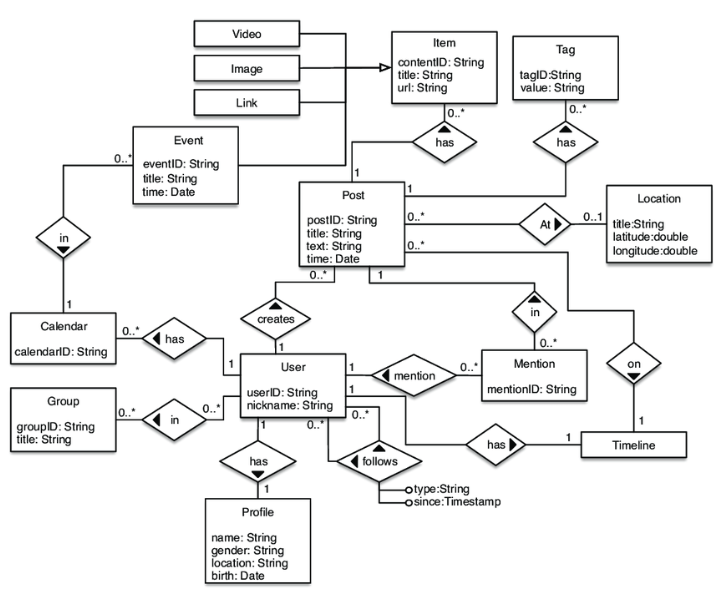
* **Object Diagram:**

****

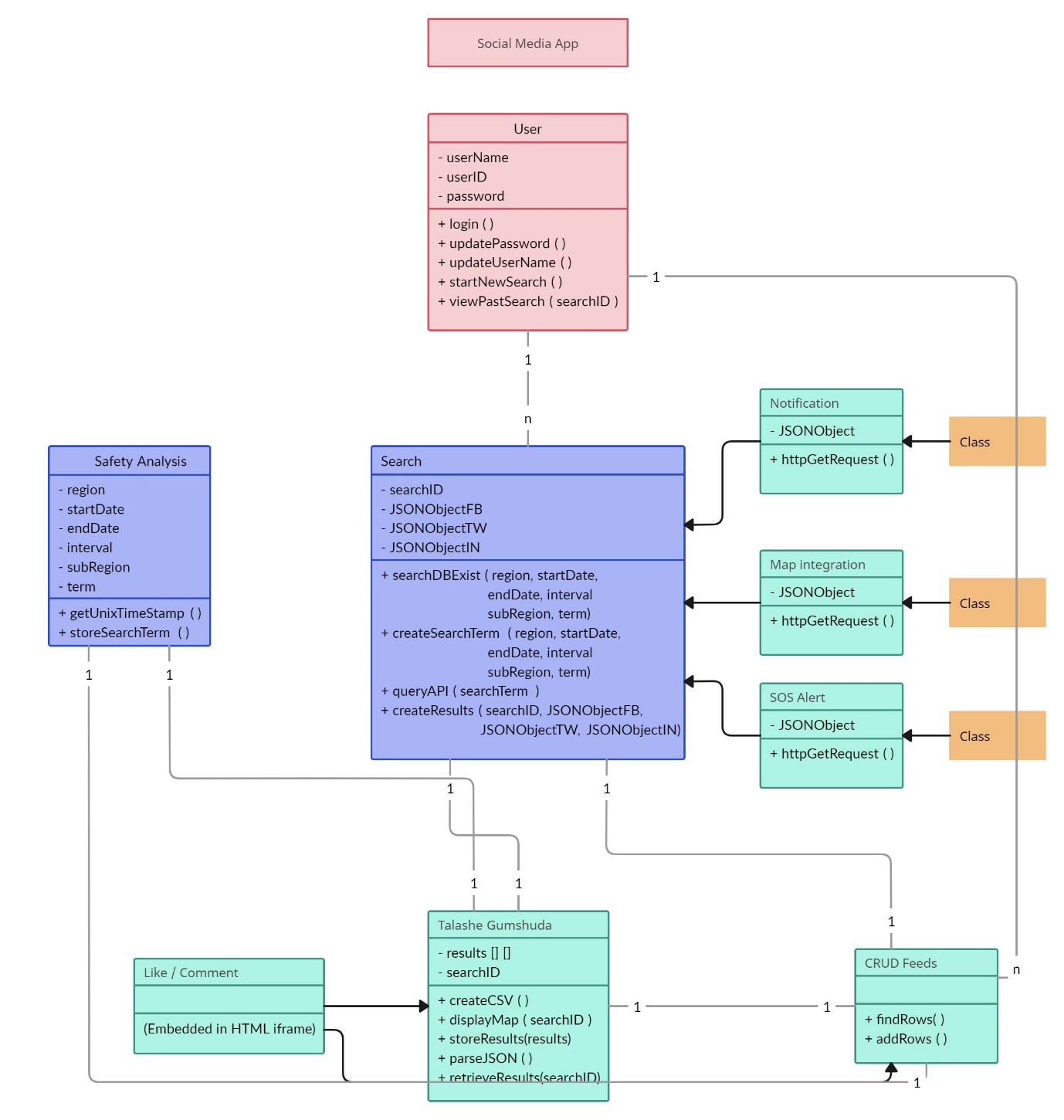
* **Collaboration Diagram**

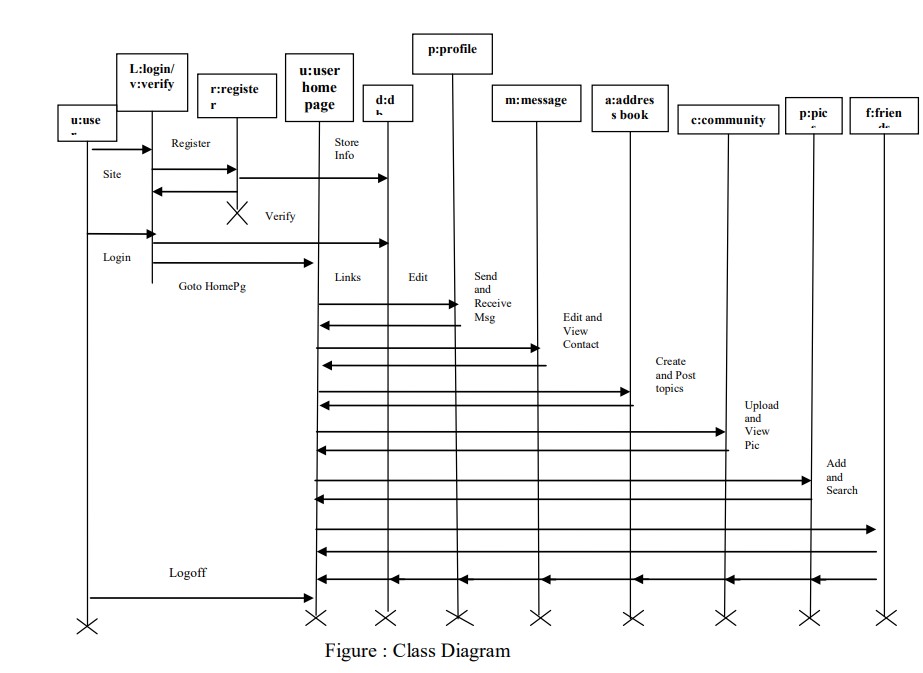
****

* **Activity Diagrams**

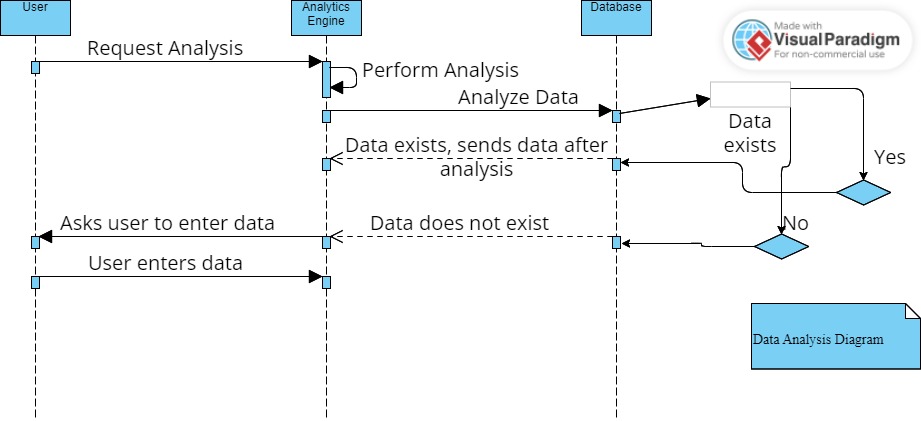
****

* **Class Diagram**

****

****

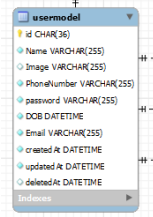
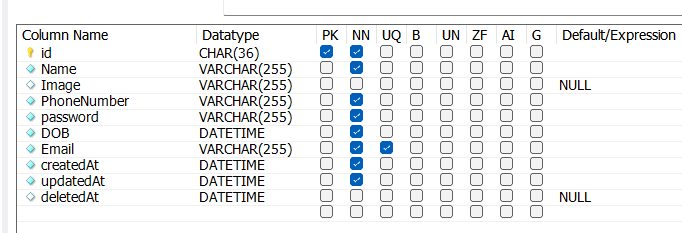
* **Analysis Diagram:**

****

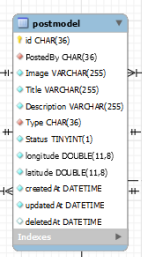
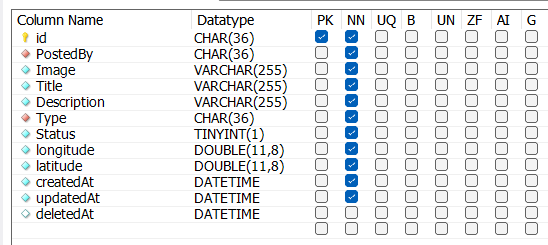
**SDS PART**

**2.3 DATA DESIGN**

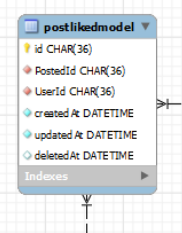
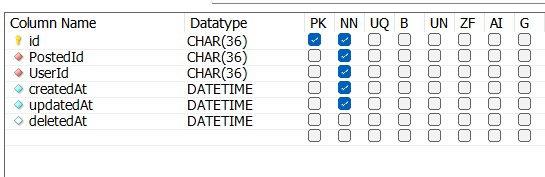
**2.3.1 USERS**



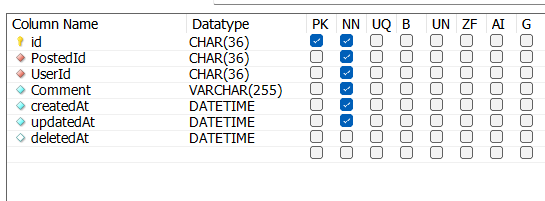
**2.3.2 POSTS**

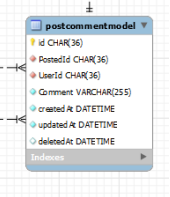


**2.3.3 POST LIKED MODEL**

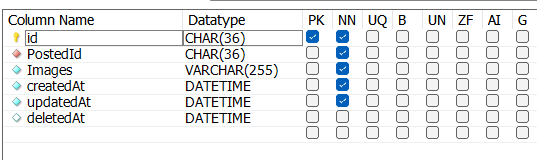


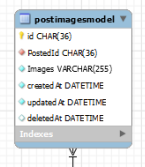
**2.3.4 POST COMMENT MODEL**



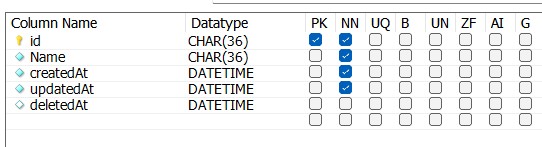


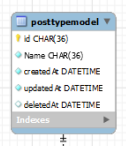
**2.3.5 POST IMAGES MODEL**



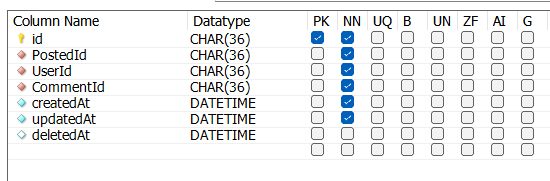


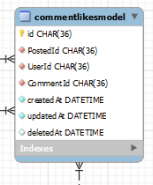
**2.3.6 POST TYPE MODEL**



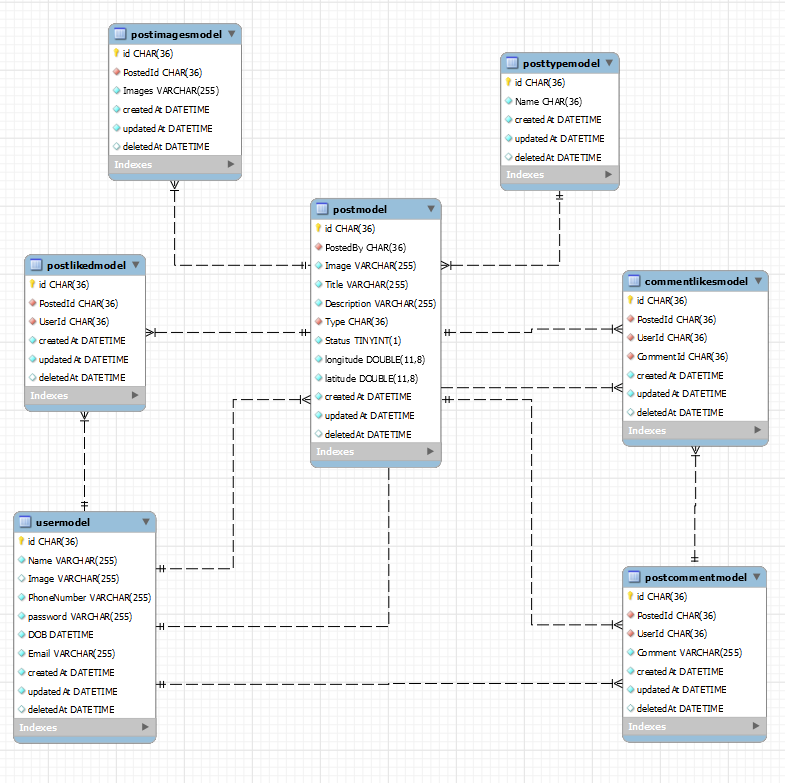


**2.3.7 Comment Likes Model**



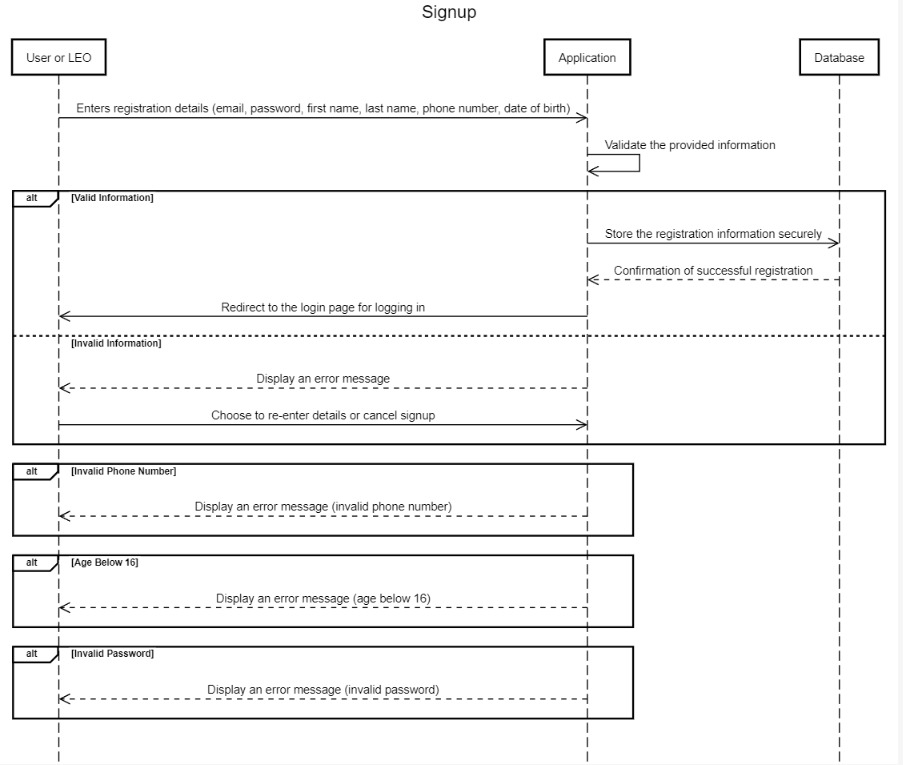


ERD

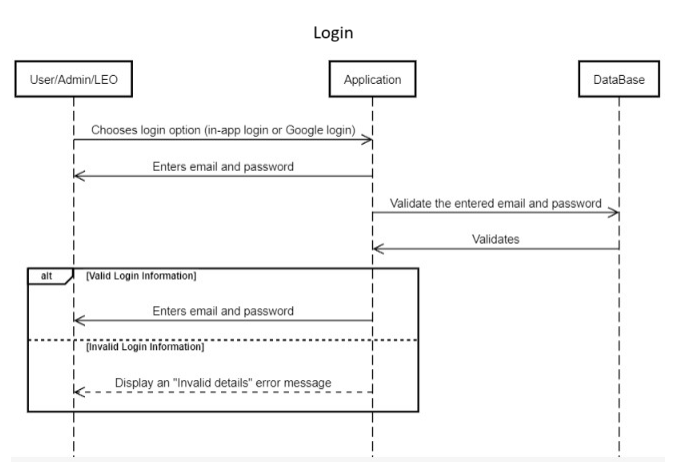


SEQUENCE DIAGRAM

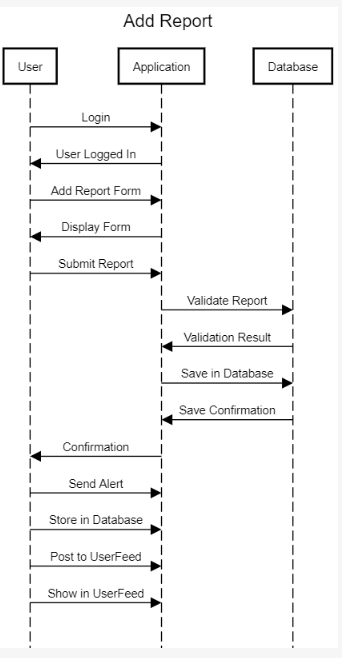
**SignUp:**



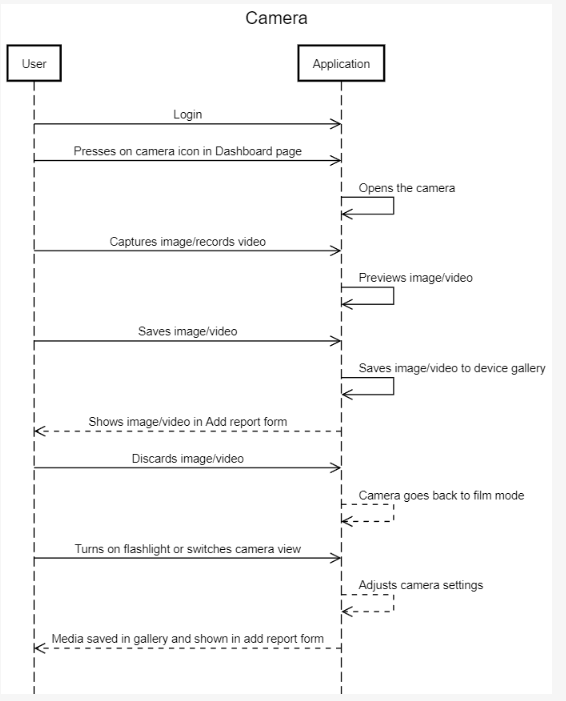
**Login:**



**Add Report:**

****

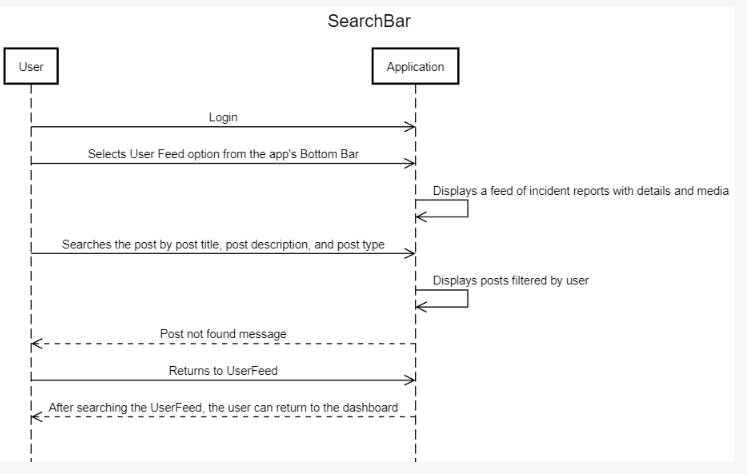
**Camera:**

****

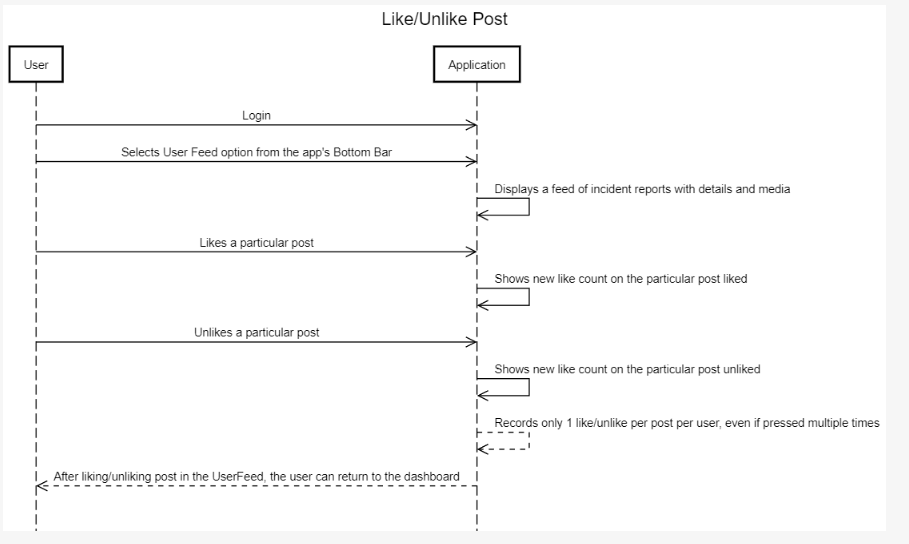
**UserFeed:**

****

**SearchBar:**

****

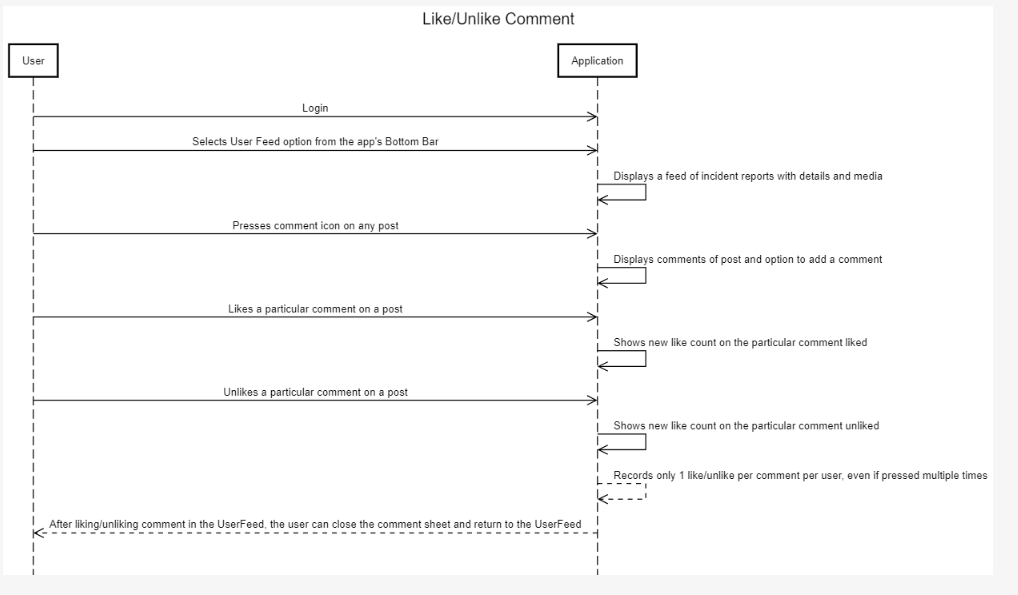
**Like/Unlike Post:**

****

**Add Comment:**

****

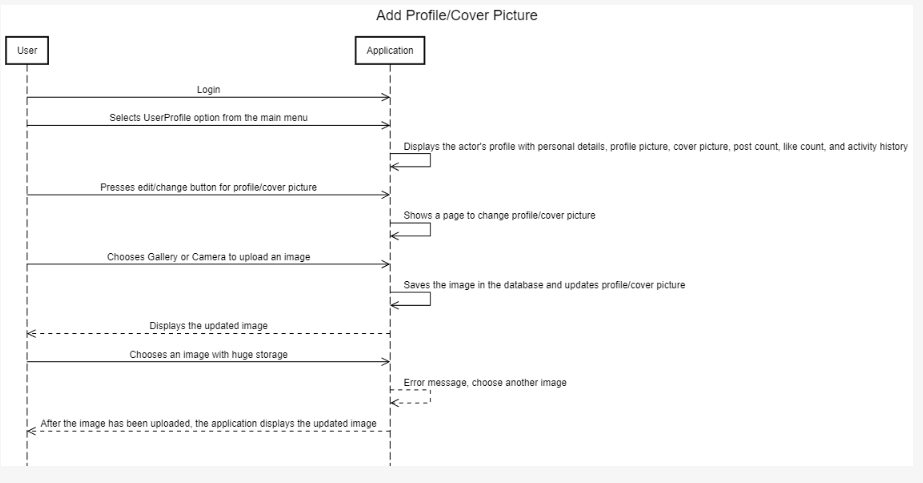
**Like/Unlike Comment:**

****

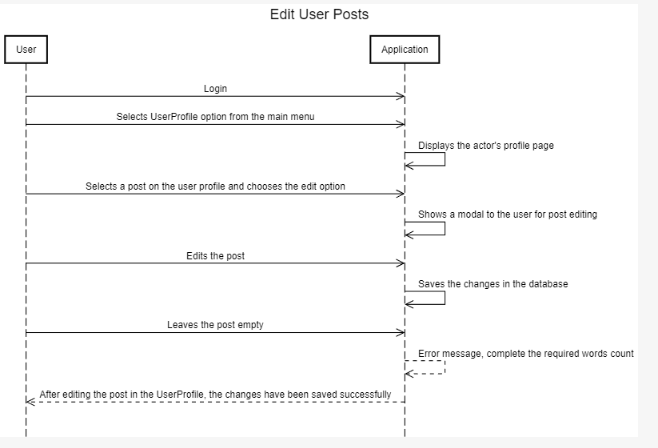
**UserProfile:**

****

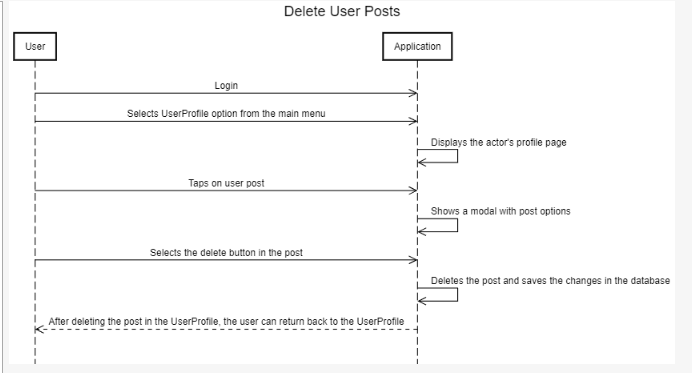
**Add Profile/ Cover Picture:**

****

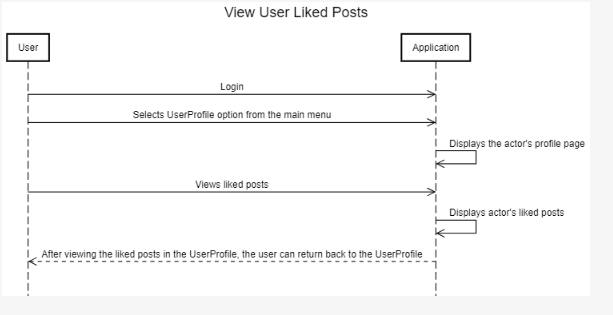
**Edit User Posts:**

****

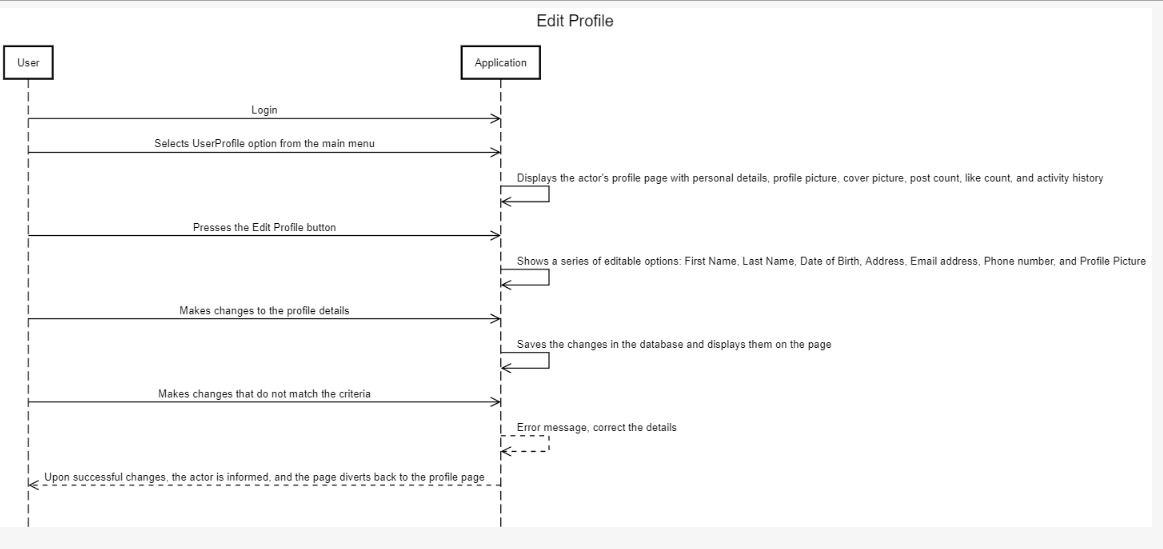
**Delete User Posts:**

****

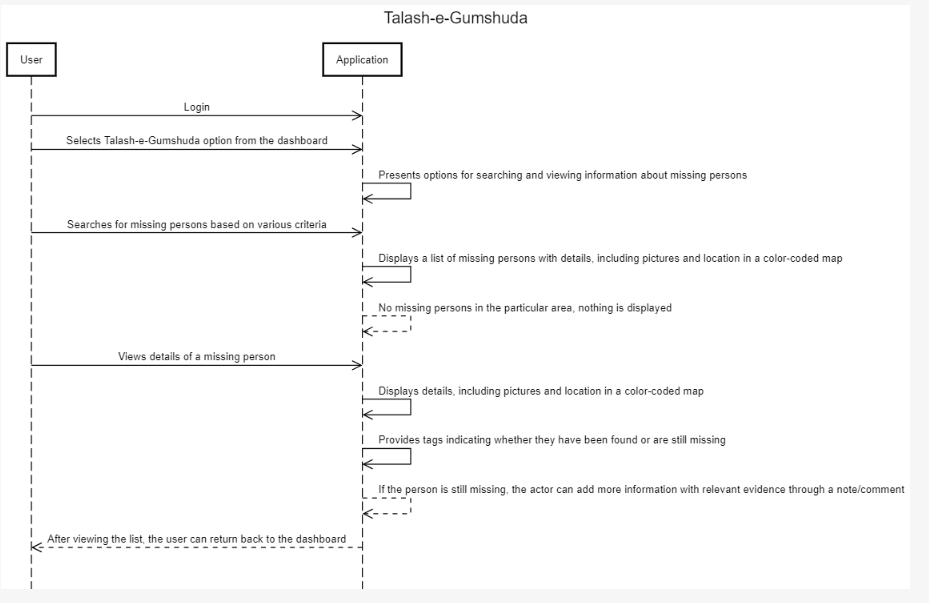
**View User Liked Posts:**

****

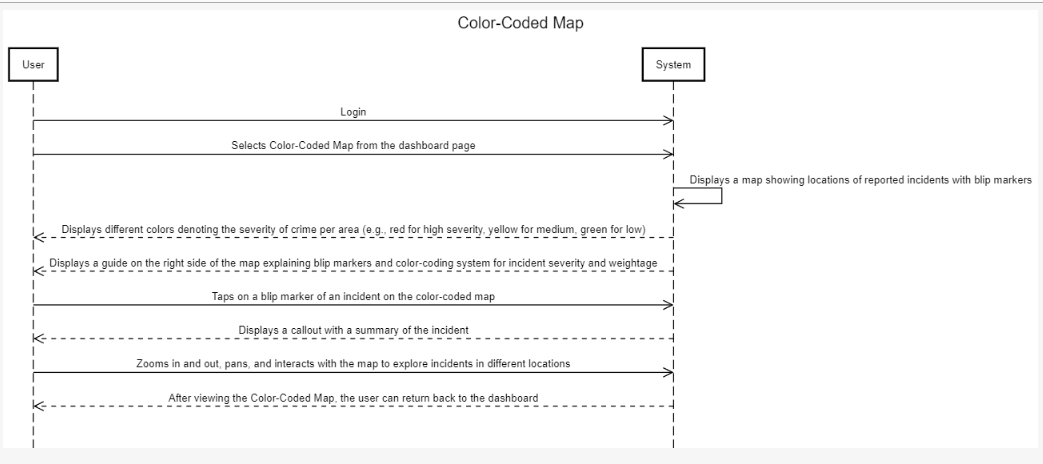
**Edit Profile:**

****

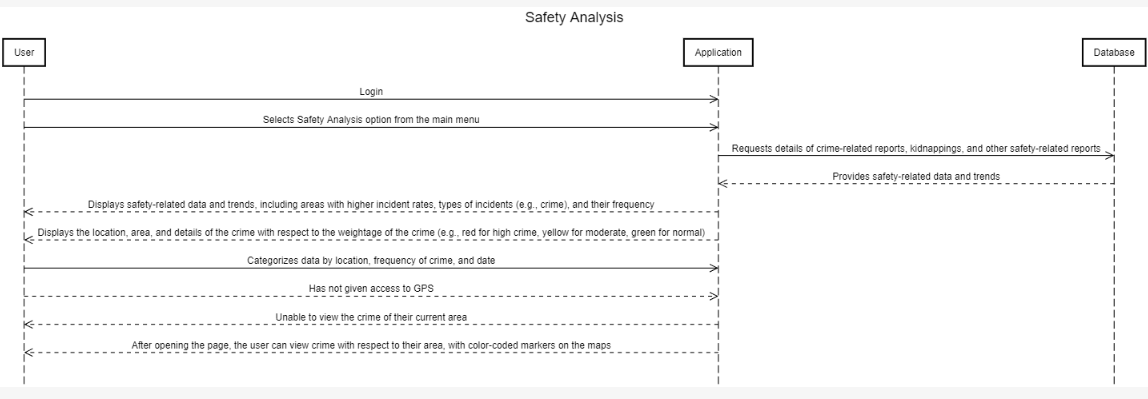
**Talash-e-Gumshuda:**

****

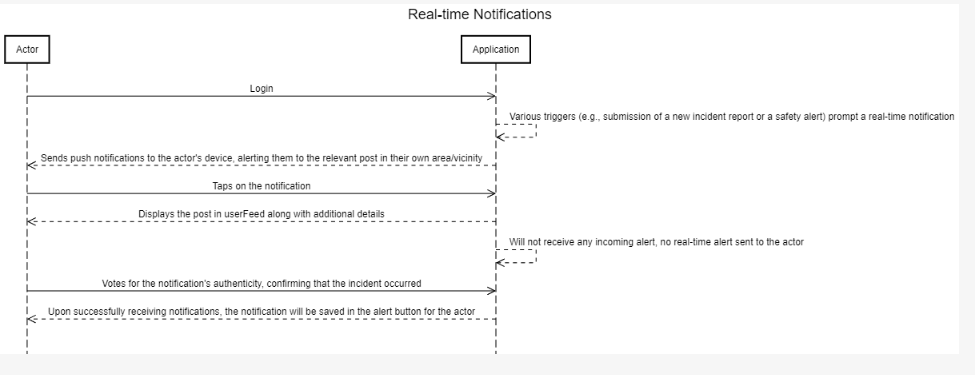
**Color-Coded Map:**

****

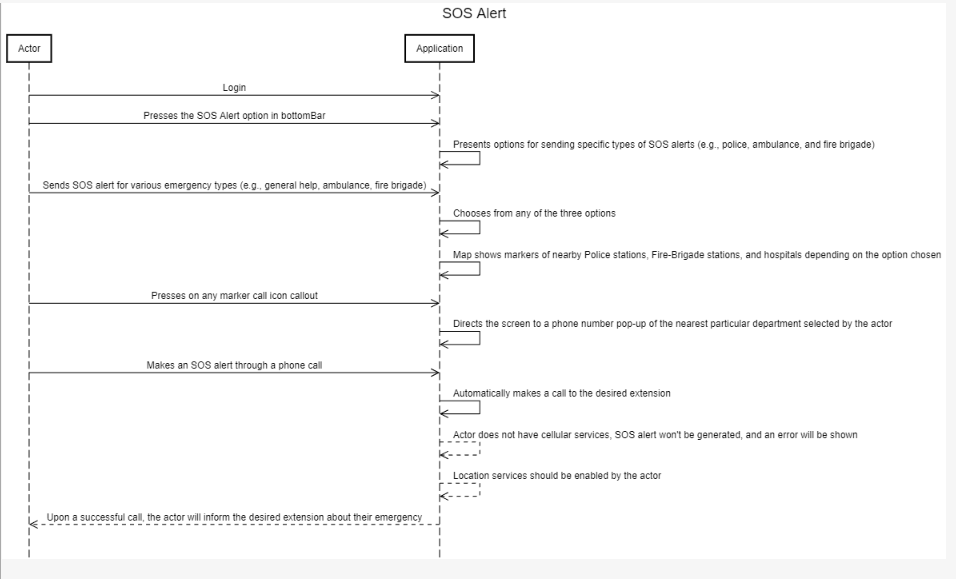
**Safety Analysis:**

****

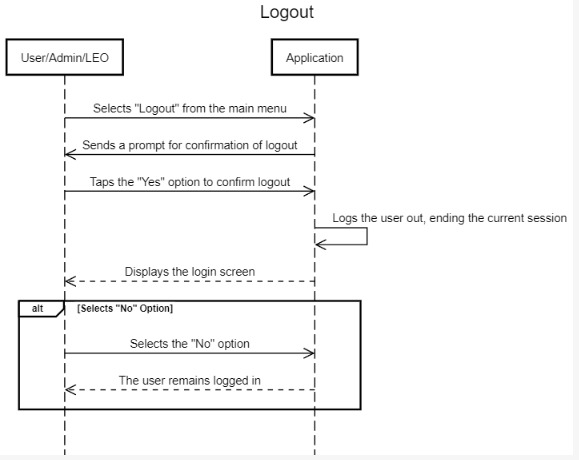
**Real-Time Notification:**

****

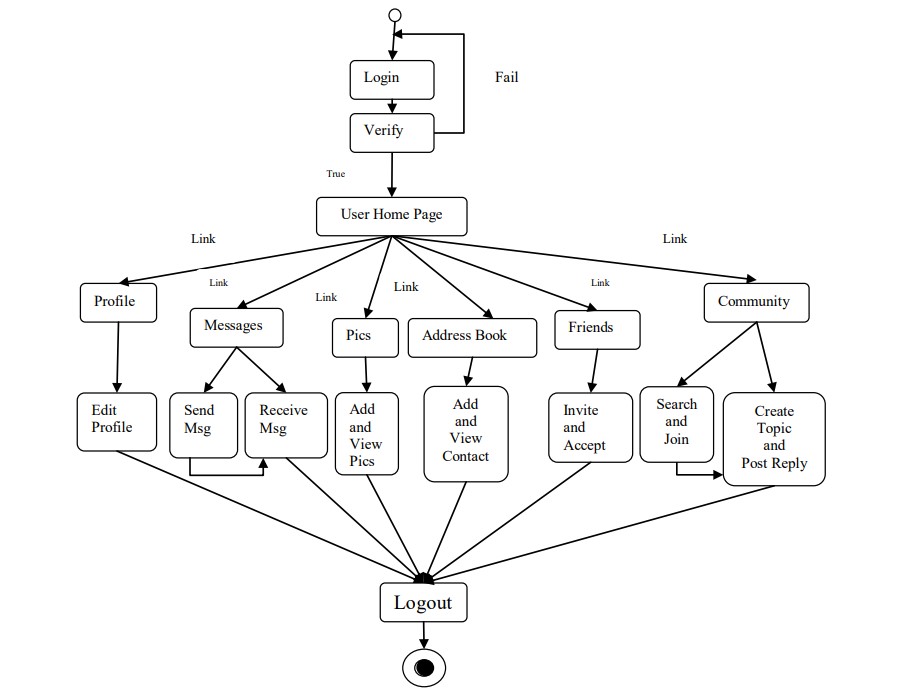
**SOS Alert:**

****

**Logout:**



Domain Model:



1. **TEST CASES**
   1. Sign up

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | Sign up | **Test Case Description** | | This test case describes how a User signs up into the Crime-Safety Application | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 21 Oct 2023 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | Must be a new User to sign up in to the Application | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Press the Get Started icon on onboarding Page | | Application redirects them and shows them the sign up page | | Application shows sign up page | | | Pass | | |
| 2 | Enters valid details | | Application should display login in page | | Application displays log in page | | | Pass | | |
| 3 | Enter invalid | | Application should not display log in page | | Application does not display log in page | | | Pass | | |

1.2 Login

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | Login | **Test Case Description** | | This test case describes how a User, admin, or LEO logs into the Crime-Safety Application | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 21 Oct 2023 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | Must be a User, admin, or LEO to log in to the Application | | |  |  |  |  |  |  |  |
| 2 | Must be registered user or LEO. | | |  |  |  |  |  |  |  |
| 3 | Admin must have valid credentials | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Choose sign in option | | Application redirects them and shows them the login page | | Applications shows Sign in Page | | | Pass | | |
| 2 | Enters valid email and password | | Application should display dashboard | | Application displays dashboard | | | Pass | | |
| 3 | Enter invalid | | Application should not display dashboard | | Application Does not display dashboard | | | Pass | | |

1.3 Add report

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | Add Report | **Test Case Description** | | This test case is about adding a report in Crime-Safety Application | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 10 Nov Oct 2023 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | The user has to be already logged in. | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Choose add report option in dashboard | | Application should show add report modal | | Application shows add report modal | | | Pass | | |
| 2 | Enters valid details along with media | | Application should display a success alert and save data into database | | Application displays a success alert and save data into database | | | Pass | | |
| 3 | Enters invalid details along with media | | Application should display an error message | | Application displays an error message | | | Pass | | |

1.4 Camera

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | Camera | **Test Case Description** | | This test case is about capturing an image or recording a video through Camera. | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 10 Nov Oct 2023 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | The user has to be already logged in. | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Choose add Camera icon in dashboard | | Application should open device camera | | Application opens device camera | | | Pass | | |
| 2 | Capture Image or record Video | | Application should Preview Image or Video | | Application Previews Image or Video | | | Pass | | |
| 3 | Save Image/Video | | Application should save image in gallery and display it in Add report form | | Application saves image in gallery and display it in Add report form | | | Pass | | |
| 4. | Discard Image or Video | | Application should take camera back to film mode | | Application takes camera back to film mode | | | Pass | | |

1.5 UserFeed

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | UserFeed | **Test Case Description** | | This test case is about UserFeed | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 12 Nov Oct 2023 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | The user has to be already logged in. | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Choose UserFeed option from Bottom Bar | | Application should display bottom bar | | Application displays bottom bar | | | Pass | | |
| 2 | Choose comment icon of a particular post | | Application should display comments of a particular post | | Application displays comments of a particular post | | | Pass | | |

1.6 SearchBar

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | Searchbar | **Test Case Description** | | This test case is about UserFeed SearchBar | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 12 Nov Oct 2023 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | The user has to be already logged in. | | |  |  |  |  |  |  |  |
| 2 | User must be in UserFeed | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Search the posts by post title, post description and post type | | Application should display posts as filtered by user | | Application displays posts as filtered by user | | | Pass | | |
| 2 | Search posts that does not exist | | Application should display post not found | | Application displays post not found | | | Pass | | |

1.7 Like/Unlike Post

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | Like/Unlike Post | **Test Case Description** | | This test case is about liking/unliking post in the userFeed | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 12 Nov Oct 2023 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | The user has to be already logged in. | | |  |  |  |  |  |  |  |
| 2 | User must be in UserFeed | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Press Like button on a particular post. | | Application should display new like count on the particular post liked and save response to database | | Application displays new like count on the particular post liked and saves response to database | | | Pass | | |
| 2 | Press unLike button on a particular post. | | Application should display new like count on the particular post unliked and save response to database | | Application displays new like count on the particular post unliked and saves response to database | | | Pass | | |

1.8 Add comment

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | Add Comment | **Test Case Description** | | This test case is about adding comment on a post in the userFeed | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 12 Nov Oct 2023 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | The user has to be already logged in. | | |  |  |  |  |  |  |  |
| 2 | User must be in UserFeed | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Press comment icon a post | | Application should display comments of a particular post | | Application displays comments of a particular post | | | Pass | | |
| 2 | Add a comment | | Application should securely stores user comment into the comment database. | | Application securely stores user comment into the comment database. | | | Pass | | |

1.9 Like/Unlike Comment

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | Like/Unlike comment | **Test Case Description** | | This test case is about liking/unliking comments in the userFeed | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 15 Nov Oct 2023 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | The user has to be already logged in. | | |  |  |  |  |  |  |  |
| 2 | User must be in UserFeed | | |  |  |  |  |  |  |  |
| 3 | User must be in comment sheet of a post | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Press Like button on a particular comment. | | Application should display new like count on the particular comment liked and save response to database | | Application displays new like count on the particular comment liked and saves response to database | | | Pass | | |
| 2 | Press unlike button on a particular comment. | | Application should display new like count on the particular comment unliked and save response to database | | Application display new like count on the particular comment unliked and save response to database | | | Pass | | |

1.10 UserProfile

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | UserProfile | **Test Case Description** | | This test case is about user’s profile | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 15 Nov Oct 2023 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | The user and admin has to be already logged in. | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Choose UserProfile option from Bottom Bar | | Application should display the user’s profile, including personal details, profile picture,cover picture , post count, like count and activity history | | Application displays the user's profile, including personal details, profile picture,cover picture , post count, like count and activity history | | | Pass | | |

1.11 Add Profile/ Cover Picture

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | Add Profile/ Cover Picture | **Test Case Description** | | This test case is about Add Profile/ Cover Picture in the userprofile | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 15 Nov Oct 2023 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | The user has to be already logged in. | | |  |  |  |  |  |  |  |
| 2 | User must be in UserProfile | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Press camera button on Profile Picture or Cover Picture. | | Application should open device gallery | | Application opens device gallery | | | Pass | | |
| 2 | Choose image | | Application should display new profile/cover picture and save image in database | | Application displays new profile/cover picture and save image in database | | | Pass | | |

1.12 Edit User Posts

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | Edit User Posts | **Test Case Description** | | This test case is about editing user’s posts in UserProfile | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 29 Nov Oct 2023 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | The user and admin have to be already logged in. | | |  |  |  |  |  |  |  |
| 2 | User must be in UserProfile | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Press on any user post. | | Application should display a modal with edit button | | Application displays a modal with edit button | | | Pass | | |
| 2 | Edit post | | Application should edit post and saves changes in user post database | | Application edits post and saves changes in user post database | | | Pass | | |

1.13 Delete User Posts

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | Delete User Posts | **Test Case Description** | | This test case is about deleting user’s posts in UserProfile | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 29 Nov Oct 2023 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | The user and admin have to be already logged in. | | |  |  |  |  |  |  |  |
| 2 | User must be in UserProfile | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Press on any user post. | | Application should display a modal with delete button | | Application displays a modal with delete button | | | Pass | | |
| 2 | Delete post | | Application should delete post and saves changes in user post database | | Application deletes post and saves changes in user post database | | | Pass | | |

1.14 View User Liked Posts

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | View User Liked Posts | **Test Case Description** | | This test case is about viewing user liked posts UserProfile | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 4 Dec 2023 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | The user has to be already logged in. | | |  |  |  |  |  |  |  |
| 2 | User must be in UserProfile | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Press on any user liked post. | | Application should display a modal with user’s liked post | | Application displays a modal with user’s liked post | | | Pass | | |

1.15 Edit Profile

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | Edit Profile | **Test Case Description** | | This test case is about Edit Profile | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 4 Dec 2023 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | The user has to be already logged in. | | |  |  |  |  |  |  |  |
| 2 | User must be in UserProfile | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Press on Edit Profile button in UserProfile | | Application should display EditProfile page | | Application display EditProfile page | | | Pass | | |
| 2 | Edit personal details with valid credentials | | Application should save changes to database and notify user | | Application saves changes to database and notify user | | | Pass | | |
| 3 | Edit personal details with invalid credentials | | Application should display an error | | Application displays an error | | | Pass | | |

1.16 Talash-e-Gumshuda

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | Talash-e-Gumshuda | **Test Case Description** | | This test case is about the missing pupil in the city | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 15 Dec 2023 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | The user has to be already logged in. | | |  |  |  |  |  |  |  |
| 2. | The user must be in Talash-e-Gumshuda page | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Search for missing persons based on various criteria, such as name, age, gender, and location | | Application should display a list of missing persons with their details, including pictures and location in color-coded map, when available | | Application display a list of missing persons with their details, including pictures and location in color-coded map, when available | | | Pass | | |
| 2 | Searches tag | | Should display tags for each person | | Displays tags for each person | | | Pass | | |

1.17 Color-Coded Map

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | Color-Coded Map | **Test Case Description** | | This test case is about the is about Color-coded Map | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 15 Dec 2023 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | The user has to be already logged in. | | |  |  |  |  |  |  |  |
| 2. | The location services should be enabled | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Interact with the Map | | Application should display a map showing the locations of reported incidents by a marker and different colors of area color coding denoting the severity of crime per area (e.g., red for high severity, yellow for medium, green for low) | | Application displays a map showing the locations of reported incidents by a marker and different colors of area color coding denoting the severity of crime per area (e.g., red for high severity, yellow for medium, green for low) | | | Pass | | |
| 2 | Tap on a Marker | | A callout should appear with summary of incident | | A callout appears with summary of incident | | | Pass | | |

1.18 Safety Analysis

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | Safety Analysis | **Test Case Description** | | This test case is about Safety Analysis | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 28 Dec 2023 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | The user has to be already logged in. | | |  |  |  |  |  |  |  |
| 2. | The user must be in Safety Analysis page | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Choose Crime Analysis by various categorization | | Application should display crime analysis by various categorization | | Application displays crime analysis by various categorization | | | Pass | | |

1.19 Real-time Notifications

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | Real-time Notifications | **Test Case Description** | | This test case is about Real-time Notification | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 28 Dec 2023 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | The user has to be already logged in. | | |  |  |  |  |  |  |  |
| 2. | The user must have notification enabled | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Tap on notification bell | | Application should display latest push notifications | | Application displays latest push notifications | | | Pass | | |

1.20 SOS alert

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | SOS Alert | **Test Case Description** | | This test case is about SOS Alert | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 5 Jan 2024 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | The user has to be already logged in. | | |  |  |  |  |  |  |  |
| 2. | The user must have location services enabled | | |  |  |  |  |  |  |  |
| 3. | The user must have cellular services | | |  |  |  |  |  |  |  |
| 4. | The user must be in SOS page | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Tap on emergency service department marker | | Application should open a callout with call option | | Application opens a callout with call option | | | Pass | | |
| 2 | Press call icon | | Should make a call through user device | | Makes a call through user device | | | Pass | | |

1.21 Logout

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Name** | | Logouut | **Test Case Description** | | This use case describes how a User, admin, or LEO logs out from the Crime-Safety Application | | | | | |
| **Created By** | | Akbar/ | **Verion** | | 1.1 | | **Date Tested** | | 21 Oct 2023 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  |  |  |  |  |  |  |
| 1 | User, LEO and Admin must be already logged in | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Choose Logout in dashboard | | Application should redirect and displays login page | | Application redirects and displays login page | | | Pass | | |

* **User Manual:**

Crime-Safety User Manual

Table of Contents

1. Introduction
2. Getting Started
   1. Onboarding
   2. Signup
   3. Login
3. Main Interface
   1. Map
   2. Markers
   3. Posts
4. User Interaction
   1. Posting
   2. Profile
   3. Edit Profile
5. Introduction

Welcome to Crime-Safety application! This app is designed to help you stay informed about your surroundings and contribute to the safety of your community.

1. Getting Started
2. **Onboarding:**

When you first launch the app, you will be guided through the onboarding process. Follow the prompts to set up your account.

1. **Signup:**

To create an account, click on the "Sign Up" button.

Provide the required information such as username, email, and password.

After filling in the details, click "Sign Up" to create your account.

1. **Login**

If you already have an account, click on the "Login" button.

Enter your username and password.

Click "Login" to access the main features of the app.

1. **Main Interface:**
2. Map

After logging in, you will be presented with a map displaying live location information.

The map will have markers indicating crime-related tags, police stations, fire brigades, and hospitals.

1. Markers

Crime-related tags are represented by specific markers on the map.

Police stations, fire brigades, and hospitals also have distinct markers.

1. Posts

Posts containing crime-related or other updates will be displayed on the map.

You can click on a post to view more details.

1. SOS:

The User can call for help by interacting with options such as Police, Ambulance/Hospital and Fire brigade.

1. User Interaction
2. Posting

To contribute to the community, click on the plus button to create a post.

Add details about the incident, upload images, and pinpoint the location on the map.

Click "Post" to share the information with the community.

1. Profile

Access your profile to view your activity and posts.

Click on a post in your profile to view or edit its details.

1. Edit Profile

Navigate to the profile section.

Click on "Edit Profile" to make changes to your account information.

Save your changes to update your profile.